PC104-30F/G

100kHz/330kHz Data Acquisition Boards

User's Manual

For PC104 Compatible Computer Systems

First Edition

August 1999

22 August 1999 Printing

Information furnished in this manual is believed to be accurate and reliable; however no responsibility is assumed for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

Table of Contents

Page

Overview 1-1
Features....................1-2
Programmable Gain....................1-2
A/D Sub-system....................1-2
Key Specifications....................1-3
D/A Sub-system....................1-3
Key Specifications: DACO thru DAC3....................1-3
Digital I/O Sub-system....................1-4
Interface logic 1-4
Software Support....................1-4
EDR Features....................1-5
Waveview Features....................1-6
Visual Basic Custom Controls Features....................1-6
Throughput....................1-7
DMA 1-7
Program Transfer 1-7
Getting Started 1-7
Accessories....................1-8
ADPT37-40 Adaptor....................1-8
26 Way 2mm to IDC26 Ribbon Convertor Cable 1-8
40 Way 2mm to IDC40 Ribbon Convertor Cable 1-8
PC-78 1-9
PC-81 1-9
PC-22 1-9
PC-68 1-9
Introduction....................2-1
D/A Sub-system....................2-1
A/D Sub-system....................2-1
Bus Interface....................2-3
Timing and Control 2-3
Crystal Oscillator 2-4
Clock Divider....................2-4
Clock Selection Multiplexer....................2-4
Counter/Timer....................2-5
A/D Operations....................2-5
Configuring the Board....................2-5
Sampling Data....................2-6
Simple Polled I/O 2-6

Single Block DMA 2-7
Multi Block DMA 2-7
Block Mode Triggering 2-8
Normal Mode..2-8
Block Trigger Mode..2-8
Operation of the Channel List..2-10
Digital I/O...2-10
Mode 0 (Basic Input/Output)..2-10
Mode 1 (Strobed Input/Output) 2-11
Mode 2 (Strobed Bi-directional Input/Output) 2-11
Introduction..3-1
Changing the Configuration...3-11
Bus Interface Configuration...3-11
Base Address 3-11
Interrupt Level 3-11
Selection of Interrupt Level..3-11
Wait State Jumper..3-12
Interrupt Source 3-13
DMA Level..3-14
Selection of DMA Levels..3-14
D/A Selections 3-14
A/D Configuration...3-14
A/D Input Mode...3-15
A/D Voltage Range Setting 3-15
A/D Clock/Trigger 3-17
Uncommitted Counter/Timer 3-17
External Clock Line 3-17
Counter/Timer Clock Source...3-17
Counter/Timer Enable 3-19
Introduction..4-1
Connections to the PC104 Connectors..4-1
Analog User Connector [AD-CON]...4-1
Signal Definitions...4-2
Digital User Connector [P1 - DIG40] 4-4
Signal Definitions...4-5
Analogue I/O..4-7
Recommended Analogue Input Schemes 4-7
Single Ended Inputs 4-7
Differential Inputs...4-7
Analogue Output..4-9
Connection Guidelines 4-9
Shielded Input Lines 4-9
Grounding...4-9
Input Voltages 4-9

Source Impedance..4-9
Digital I/O...4-10
Mode 0 (Basic Input/Output)..4-10
Mode 1 (Strobed Input/Output)..4-11
Handshake Signals..4-12
Mode 2 (Strobed Bi-directional Input/Output)....................................4-13
Handshake Signals..4-14
Mode Combination Considerations 4-15
Connecting Normally Open devices to the PPI 4-15
Introduction..5-1
Register Structure..5-1
ADDATL - A/D Data Low Byte (Offset 0) (Read Only) 5-4
BLKCNT - Block Counter (Offset 0) 5-4
ADDSR - A/D Data/Status Register (Offset 1)..5-5
ADCCR - A/D Control/Channel Register (Offset 2)..................................5-7
ADMDE - A/D Mode Register (Offset 3) 5-8
PRESCALER - A/D Clock Prescaler Register (Offset 4)........................5-12
DIVIDER - A/D Clock Divider Register (Offset 5)................................5-13
USR_CNT - User Counter Register (Offset 6) 5-14
TMRCTR - Timer Control Register (Offset 7) 5-15
DIOP0 - Digital I/O Port 0 (Offset 8)..5-19
DIOP1 - Digital I/O Port 1 (Offset 9)..5-19
DIOP2 - Digital I/O Port 2 (Offset 10)...5-20
DIOCNTRL - Digital I/O Control (Offset 11) 5-21
Configuration Mode...5-22
Bit Set/Reset Mode..5-23
DADATL0 - DAC0 Register (Low Byte) (Offset 12)..............................5-23
DADATH0 - DAC0 Register High Byte (Offset 13)................................5-24
DADATL1 -DAC1 Register (Low Byte) (Offset 16)................................5-24
DADATH1 - DAC1 Register High Byte (Offset 17)................................5-25
IMUX – DMA/IRQ Config Register (Offset 18)......................................5-25
IGATE – DACK/Gate Control Register (Offset 19) 5-27
DADATL2 –DAC2 Register (Low Byte) (Offset 20) 5-31
DADATH2 – DAC2 Register High Byte (Offset 21)................................5-31
DADATL3 -DAC1 Register (Low Byte) (Offset 22).................................5-31
DADATH3 – DAC3 Register High Byte (Offset 23)................................5-32
GAINREG - Gain Read Back and Board Type Register (Offset 24)........5-32
GMEM0 - Gain Memory 0 Register (Offset 24).......................................5-35
GMEM1 - Gain Memory 1 Register (Offset 25).......................................5-36
GMEM2 - Gain Memory 2 Register (Offset 26).......................................5-37
GMEM3 - Gain Memory 3 Register (Offset 27).......................................5-38
Configuration Registers...5-39
ADCCFG - A/D Configuration Register (Offset 28)................................5-39
DACCFG - D/A Configuration Register (Offset 29)................................5-41

CLKSRC - Clock Source Configuration Register (Offset 30) 5-41
Introduction..6-1
Converting From Binary To Analogue Values 6-1
A/D Converter Codes..6-1
DAC0 and DAC1 Converter Codes...6-1
DAC2 and DAC3 Converter Codes...6-2
Initialisation 6-2
Clearing the A/D Subsystem..6-3
Writing to the D/A Converters...6-4
Digital I/O 6-4
Setting Channel Gain...6-4
Obtaining a Single A/D Reading..6-5
Setting the Sample Rate..6-5
Loading the Channel List/Block Counter 6-6
Obtaining a Series of A/D Conversions by Polled I/O...6-7
Interrupts..6-7
Single Channel DMA 6-8
Dual Channel Gap-free DMA...6-9
DMA Data Format 6-11
Dealing With Extended Memory 6-11
Error Detection..6-12
End of DMA Block Interrupt...6-12
Introduction...7-1
A/D Calibration..7-1
Requirements 7-1
Equipment Required..7-2
Setting the Reference Voltage 7-2
+10V 7-2
Calibration Procedure...7-3
A/D Calibration for the PC104-30G/PC104-30GA Boards.................7-3
A/D Calibration for the PC104-30F/PC104-30FA Boards..................7-3
A/D Calibration Software..7-5
DAC0 to DAC3 Calibration 7-5
Introduction...9-1
New Series Boards..9-1
Running Old Software on the PC-30F and G 9-1
Running PC104-30F and G Software on Older "New Series" Boards 9-1
Introduction..11-1
The Diagnostics Function 11-1
Common Problems..11-2
Waveview Cannot Find the Board...11-2
A/D Output Code All Zeros or All Ones...11-2
A/D Readings Are Noisy...11-2
First Reading in a Series is Inaccurate 11-2

The Board Does Not Operate at Full Throughput 11-3
Board Does Not Operate in Block Trigger Mode 11-3
'Board not Found' Message Appears when Running Test Software (ie. DEMO Programs, Waveview etc.)..11-3
Readings from Channel 0 through 15 are Erratic and Unstable 11-4
Sampled Data Via Single or Dual Channel DMA are Erroneous.............11-4
DMA and Polled I/O Voltage Mismatch...11-7
Sampled Data Indicates Excessive Noise Levels 11-7
DAC Outputs clips to ±13V irrespective of DAC Input Data..................11-7
Digital I/O lines does not read the correct values on the Ports when configured as Inputs. 11-10
Pushbutton Interface to Digital Port gives random readings. 11-10
Introduction..12-2
Using Waveview 12-2
Running Waveview 12-2
The User Interface 12-3
Getting Started..12-4
Sampling Data...12-4
The Streamer...12-4
Streaming to Memory 12-5
Streaming to Disk 12-5
Continuous Buffer Streaming...12-6
Parallel Board Streaming 12-6
Program Transfer Sampling...12-7
Burst Mode..12-7
The Digital Storage Oscilloscope 12-8
The Strip Chart 12-8
The Voltmeter..12-9
Displaying Data...12-9
Controlling the Time Axis on Waveform Graphs...................................12-9
Displaying Data From Multiple Boards 12-9
Digital IO and DAC Support 12-9
Data Output, Printing and Plotting..12-10
Exporting Data As Text...12-10
Printing on Epson Compatible Dot Matrix and HP Lasetjet III Printers..12-10
Plotting on HP Plotters Or the HP Laserjet III 12-10
Getting the Most Out of Waveview Using Non-DMA Boards 12-10
Board Capabilities..12-11

List of Illustrations

Page

Figure 2 - 1. PC104-30 Board Block Diagram 2-2
Figure 2 - 2. Timing and Control Block Diagram 2-4
Figure 2 - 3. Normal Mode Operation...2-8
Figure 2 - 4. Block Mode Operation 2-9
Figure 3 - 1. PC104-30 Wait State Jumper...3-12
Figure 3 - 2. PC104-30G/GA Span Jumper...3-15
Figure 3 - 3. PC104-30 Clock Generation Schematic..................................3-17
Figure 4 - 1. PC104-30 Analog Connector..4-2
Figure 4 - 2. PC104-30 Digital Connector 4-4
Figure 4 - 3. Single Ended Analogue Inputs..4-8
Figure 4 - 4. Differential Analogue Inputs...4-8
Figure 4 - 5. Mode 0 Digital I/0...4-11
Figure 4 - 6. Mode 1 Digital Input 4-12
Figure 4 - 7. Mode 1 Digital Output..4-13
Figure 4 - 8. Mode 2 Digital I/0..4-14
Figure 4 - 9. Connecting Normally Open Devices to the PPI.......................4-15
Figure 5 - 1. A/D Data Register (Low Byte) 5-4
Figure 5 - 2. Block Count Register 5-4
Figure 5 - 3. A/D Data/Status Register 5-5
Figure 5 - 4. A/D Control/Channel Register..5-7
Figure 5 - 5. A/D Mode Register 5-9
Figure 5 - 6. Prescaler Register...5-12
Figure 5 - 7. A/D Clock Divider Register 5-13
Figure 5 - 8. User Counter Register 5-14
Figure 5 - 9. Timer Controller Register..5-15
Figure 5 - 10. Digital I/O Port 0 Register..5-19
Figure 5 - 11. Digital I/O Port 1 Register..5-19
Figure 5 - 12. Digital I/O Port 2 Register..5-20
Figure 5 - 13. Digital I/0 Control Register 5-21
Figure 5 - 14. DAC0 Low Byte Data Register 5-23
Figure 5 - 15. DAC0 High Byte Data Register..5-24
Figure 5 - 16. DACl Low Byte Data Register 5-24
Figure 5 - 17. DACl High Byte Data Register...5-25
Figure 5 - 18. A/D Data/Status Register 5-25
Figure 5 - 19. A/D Data/Status Register 5-27
Figure 5 - 20. DAC2 Low Byte Data Register 5-31
Figure 5 - 21. DAC2 High Byte Data Register..5-31
Figure 5 - 22. DAC3 Low Byte Data Register 5-32

Figure 5 - 23. DAC3 High Byte Data Register...5-32
Figure 5 - 24. GAINREG Register..5-34
Figure 5 - 25. GMEM0 Register...5-35
Figure 5 - 26. GMEM1 Register...5-36
Figure 5 - 27. GMEM2 Register...5-37
Figure 5 - 28. GMEM3 Register...5-38
Figure 5 - 29. ADCCFG Register 5-39
Figure 5 - 30. DACCFG Register 5-41
Figure 5 - 31. CLKSRC Register..5-42
Figure 7 - 1. A/D Calibration Connections..7-2
Figure B - 32. PC104-30F/G Template 10-1

Chapter 1: Introduction

Overview

The PC104-30F and G series boards are low cost, high accuracy analogue and digital I/O boards for the PC104 compatible series of computers. New ASIC technology allows many of the PC104-30 features to be controlled from software, reducing the number of jumpers on the board. The PC104-30 is a development from the well known PC30F/G Series family, and is fully compatible with these boards.

The PC104-30F and G boards are available in 4 different models as shown below:

All these boards have 16 A/D inputs and 24 programmable digital IO lines	330K A/D	100K A/D
16 SE or 8 differential inputs Programmable gains: 1, 10, 100 and 1000	PC104-30FA (4 DACS) PC104-30F (no DACS)	PC104-30GA (4 DACS) PC104-30G (no DACS)

Features

The PC104-30 can be plugged into any stackable PC104 system.

Programmable Gain

The PC104-30F, FA, G and GA models feature software programmable gain. The gain of each of the 16 input channels can be independently set to 1, 10, 100 or 1000. The gain for each channel is stored in the PC104-30's internal gain memory. In addition, the boards instrumentation amplifier can be configured to provide differential inputs from software.

A/D Sub-system

The A/D sub-system's major component is a monolithic analogue to digital converter, which accepts analogue voltage inputs from sensors, such as pressure transducers and thermocouples, and converts them into 12 bit digital codes.

This code is transmitted to the host processor, which processes it according to the software in use at the time.

The A/D section allows for 16 single-ended or 8 differential inputs, and can be configured for unipolar (input range of 0 to 10V) or bipolar (input ranges of ±5V and ±10V) operation. Resolution is 12 bits. For unipolar inputs, the output code is straight binary, and for bipolar, offset binary. The 330K models do not support unipolar inputs. On the 330K models, the input range is controlled by a bit in a register. On the 100kHz models, it is controlled by a jumper and software.

The A/D may be operated in either single conversion or continuous conversion mode. In single conversion mode the board performs a single conversion on the selected input channel and stops on completion of this conversion. In continuous conversion mode, conversions are performed at a set rate. This rate is set by programming the PC104-30's internal timer or an external clock source.

The PC104-30 contains logic which allows any sequence of channels, up to a sequence length of 31, to be selected and sampled under hardware control. This allows full throughput to be achieved even when converting multiple input channels.

A/D conversions may be monitored by either polled I/O, Direct Memory Access (DMA) or by interrupts. In polled I/O mode, the software continuously polls the board's status register to check for completion of the current A/D conversion. DMA is used to transfer data directly from the A/D to memory. In interrupt mode, the board automatically generates a hardware interrupt on completion of each conversion.

Key Specifications

- A/D resolution: 12 Bits.
- Non-linearity: Less than ∀0.75 LSB.
- A/D full scale input ranges: 0 to + 10V (GA and G models only), -5 to +5V and -10 to +10V (G, GA, F and FA models only).
- Number of A/D inputs: 16 single ended or 8 differential (F, G, FA, GA models).
- A/D throughput rate: 100 kHz or 330 kHz.

D/A Sub-system

The D/A sub-system consists of a quad 12-bit D/A converter, all configured as 12 bit D/A converters. Digital outputs are received from the host processor and converted to an analogue voltage output required by the application in hand. The four DACs are independent of one another, and can operate at a throughput of up to 130 kHz. Output ranges are independently configurable as either 0 to +10V unipolar, ±10V bipolar or ±5V bipolar from software.

Key Specifications: DACO thru DAC3

- D/A resolution: 12 Bits.
- D/A non-linearity: Within 0.01% FSR.
- Full scale output ranges: 0 to +10V, -10 to +10V, -5 to +5V.
- D/A throughput rate: 130 kHz.

Digital I/O Sub-system

The Digital I/O sub-system is an interface for the transfer of digital data from and to the PC bus to and from one or more peripheral device/s connected to the PC104-30. AN 8255 compatible chip provides three bi-directional eight bit digital I/O ports, which can each be used in a variety of operating modes.

Interface logic

The PC104-30 is accessed via I/O operations performed by the host processor. Of the 13 bit address received by the board, the most significant 8 bits select the board, and the least significant 5 bits select the register to be accessed.

The PC104-30 occupies 32 byte locations: six byte locations for the A/D sub-system, six for the D/A sub-system, four for the Digital I/O sub-system, four for the counter/timer system, and twelve for control and manufacturing test functions. The base address of the board can be selected to be located anywhere between 0000 (hex) and 1FE0 (hex).

The PC104-30 operates from the +5V line of the PC104 bus.

Software Support

The PC104-30 boards are supplied with our EDR Software Developers kit, Waveview for DOS and drivers for LabView, Matlab, Test Point and DASYLab. As new third party packages are released, we will develop drivers for them.

We also supply our "old" drivers with source code in the \EDR\OLD directory. **DO NOT** use these, as they have been superseded by EDR. They have only been included as a low level source code reference. Also remember that these drivers were written for the older boards and have not been updated for the new F and G series boards. This is not a problem as the new boards are register compatible with the old boards.

In addition to these free packages, we also sell a set of Visual Basic VBX controls. These controls can be used from any language that supports VBX controls.

EDR Features

- Supports DOS, Windows ‘95/’98/NT4
- Use C/C++, Delphi, Borland/Turbo Pascal, Quick Basic, Visual Basic etc.
- Windows and DOS streaming to disk at 330 kHz
- Complete windows streamer program written in Delphi is supplied with source code
- Circular buffer data acquisition
- Acquire data in the background
- Status function provides a “samples acquired so far” count for background operations allowing processing to be overlapped with acquisition
- Streaming into XMS under DOS
- Multiple boards can be used simultaneously to create 32 and 48 channel systems
- Extensive A/D, D/A and DIO functions
- More than 40 different error codes make it easy to debug programs
- The DOS and Windows versions provide the same API
- Conversion functions make converting between binary codes and voltages simple
- Thermocouple linearisation and RTD functions supplied
- Printed manuals and online help (Windows help format) supplied
- Free technical support is available via fax, E-mail and World Wide Web
- EDR version 1.30 supports the following boards:
- PC26, PC104-30Fx, PC104-30Gx, PC126, PC127, PC166x, PC266, PC66x, PC63x, PC36x, PC14x, PC192x and PC73x.

Waveview Features

- High speed streaming to disk
- Unique parallel board streaming
- Continuous buffer streaming
- Automatic use of XMS and EMS
- Digital storage oscilloscope
- Strip chart
- FFT’s with various window functions
- Waveform graphs of huge streamer files
- Power spectrum graphs
- Infinite pan and zoom
- Text, Epson and Laserjet III/4x/5x/6x output
- Hercules, EGA and VGA support
- Runs on XTs, ATs, 386, 486, 586, Pentium I/II,III, Celeron Processors
- Menu driven keyboard and mouse interface with online manual
- Runs under DOS 3.3 or later

Visual Basic Custom Controls Features

- EDR. The board control manages board configuration information.
- WlN. The Waveln control does high speed waveform input and streaming to disk.
- DIO. The DIO control does digital input and output.
- AD. The AtoD control does simple voltmeter type analogue to digital input.
- ISR. The ISR control allows Visual Basic programs to respond to interrupts.
- CT. The CT control programs counter/timer channels.
- DA. The DtoA control does simple digital to analogue output.

Throughput

The throughput of the PC104-30 series of boards is dependent on several factors, principally whether DMA or program transfer techniques are used to read data from the A/D converter.

DMA

DMA is Direct Memory Access and, as the name implies, data from the A/D is transferred to the PC's memory directly, without the data acquisition software in use taking any action (other than setting the hardware up initially, and waiting for the DMA to complete). In this case, the processing power of the host PC system is of no consequence, and the throughput of the PC104-30 will be at its maximum.

Program Transfer

If program transfer techniques are used (polled I/O or interrupts), the situation becomes more complex. In this case the maximum possible throughput is limited by the processing power of the CPU in the host PC, and the efficiency of the software in use. Generally, throughput of greater than 30 kHz is seldom achieved.

Note also that the throughput of the PC-30 is unaffected by the number of channels which are being converted, as long as the sequence of channels is less than 31. Longer sequences cannot use DMA, and must use program transfer techniques.

Getting Started

If you want to get started quickly and have not changed any of the factory settings on the PC104-30, here's what to do:

1. Install the PC104-30 in your computer. (Chapter 3 provides brief instructions on this, but if you are not sure, it is better to get someone who is qualified to do this).
2. Install the EDR software on the EDR CD under Windows '95/98/NT4.
3. Connect a voltage source to any (or all) of the input channels (you can also loop the analogue outputs back to the inputs). See Figure 4-1 in Chapter 4 of this manual for the pin details of the PC-30 connector.

4. Run Waveview (C:\PC104-30\Waveview\WV.exe) and go to the card menu (press <ALT+C>). Your board should be listed next to the card type option. It if is not, you should be able to select it using this option.
5. If Waveview is unable to detect your board you may have a conflict with some other hardware (eg. Network cards etc.). Refer to Chapter 3 and reconfigure your board.
6. You can use the options on Waveview Ad In menu to sample data. Pressing F8 will bring up a simple voltmeter.

Accessories

In order to assist in applying the PC104-30, several accessories are available. Only a brief description is given here. Consult your dealer for full details.

ADPT37-40 Adaptor

The ADPT37-40 Adaptor is a multi-interface adapter with and IDC40 Male Header + DB37 Female Connector for Cable Interfacing and 41 Screw Terminal Blocks for easy signal connections. It allows:

- Direct Interface to an IDC40 Ribbon Cable
- Direct Interface to a DB37 Male Cable.

26 Way 2mm to IDC26 Ribbon Convertor Cable

This cable maps a 2mm 26 IDC Female Connector on one side and the other side contains a standard 26 Way Female Connector for easy connection to Screw Terminal Adaptor Board.

40 Way 2mm to IDC40 Ribbon Convertor Cable

This cable maps a 2mm 40way IDC Female Connector on one side and the other side contains a standard 40 Way Female Connector for easy connection to Screw Terminal Adaptor Board like the ADPT37-40.

PC-78

The PC-78 is a multi-interface adapter which allows the PC104-30 to be connected to as many industry standard signal conditioning devices as possible, using only ribbon cable and standard insulation displacement connectors. It allows direct interface to:

- PC-77 screw terminal board.
- 3B series analogue signal conditioning backplanes. Up to 16 isolated signal conditioning modules are supported.
- 5B series analogue signal conditioning backplanes. Both standard and multiplexed backplanes are supported. Using the multiplexed backplanes, up 64 input and 64 output modules can be connected to a single PC104-30. 5B series modules are also isolated.
- Industry standard digital I/O backplanes. The PC-78 can be connected directly to any digital signal conditioning panel which uses standard 24 line/50-way cable type connectors. Panels are available from a variety of vendors, supporting solid state relays, isolated input modules etc.

PC-81

The PC-81 is an input expander board. Multiple PC-81's may be used to expand the input channel capability of the PC104-30 to more than 65 000 channels. Each PC-81 has 64 screw terminal inputs. Note that only Polled I/O method of sampling can be achieved if a PC-81 is used.

PC-22

The PC-22 is a Euro-card format single channel signal conditioning module. It provides programmable gain, and filtering functions.

PC-68

The PC-68 is a Euro-card format, four channel strain gage, signal conditioning board. It provides four independent channels with user programmable excitation, differential inputs, and a high performance instrumentation amplifier. The PC-68 can also be used as a four channel ultra-high performance instrumentation amplifier board.

Chapter 2: Architecture

Introduction

This chapter describes the architecture of the PC104-30 series of boards. The block diagram in Figure 2 - 1 highlights the major elements contained on the board, and their interrelationship. There are four major subsections, as follows:

D/A Sub-system

The D/A sub-system contains four 12 bit DACs as well as associated circuitry. It allows ranges: ±10V, ±5V, 0 to 10V and 0 to 13V. The Range setting are all software controlled.

A/D Sub-system

The A/D sub-system contains several separate components:

- **The Input Multiplexer**. The multiplexer selects one of 16 single ended or 8 differential input channels. This channel is selected by a channel address, obtained from the channel list. The channel list contains a list of channels to be converted and may be up to 31 channels in length. When the end of the list is reached, the A/D loops back to the first channel in the list. This channel list may also be disabled to enable compatibility with older products.
- **The Programmable Gain Amplifier**. The instrumentation/programmable gain amplifier amplifies the signal from the multiplexer by one of the four programmable gains. The gain is automatically selected by the gain code stored in the board's gain memory.
- **The Sample and Hold Unit**. The sample and hold unit holds the selected input channel steady for the duration of the A/D converter's conversion process.
- **The A/D Converter**. The A/D converter performs the actual A/D conversion. An A/D conversion is begun by an A/D strobe that is generated by the timing and control section, described later in this chapter (see under *Timing and Control* on page 2-3 of this chapter).

-
- **The FIFO buffer**. The FIFO (First In First Out) buffer is a temporary store for converted results. The FIFO stores data while the PC is performing other functions, such as memory refresh, and also stores data while the CPU changes memory buffers. Changing memory buffers allows the PC104-30 to perform DMA into the entire memory space of the PC without break. This is discussed in detail later in the chapter. The FIFO can store up to 16 samples.

Data may be transferred from A/D either by polled I/O or DMA. This is discussed later on in this chapter under the heading *A/D Operations* on page 2-5.

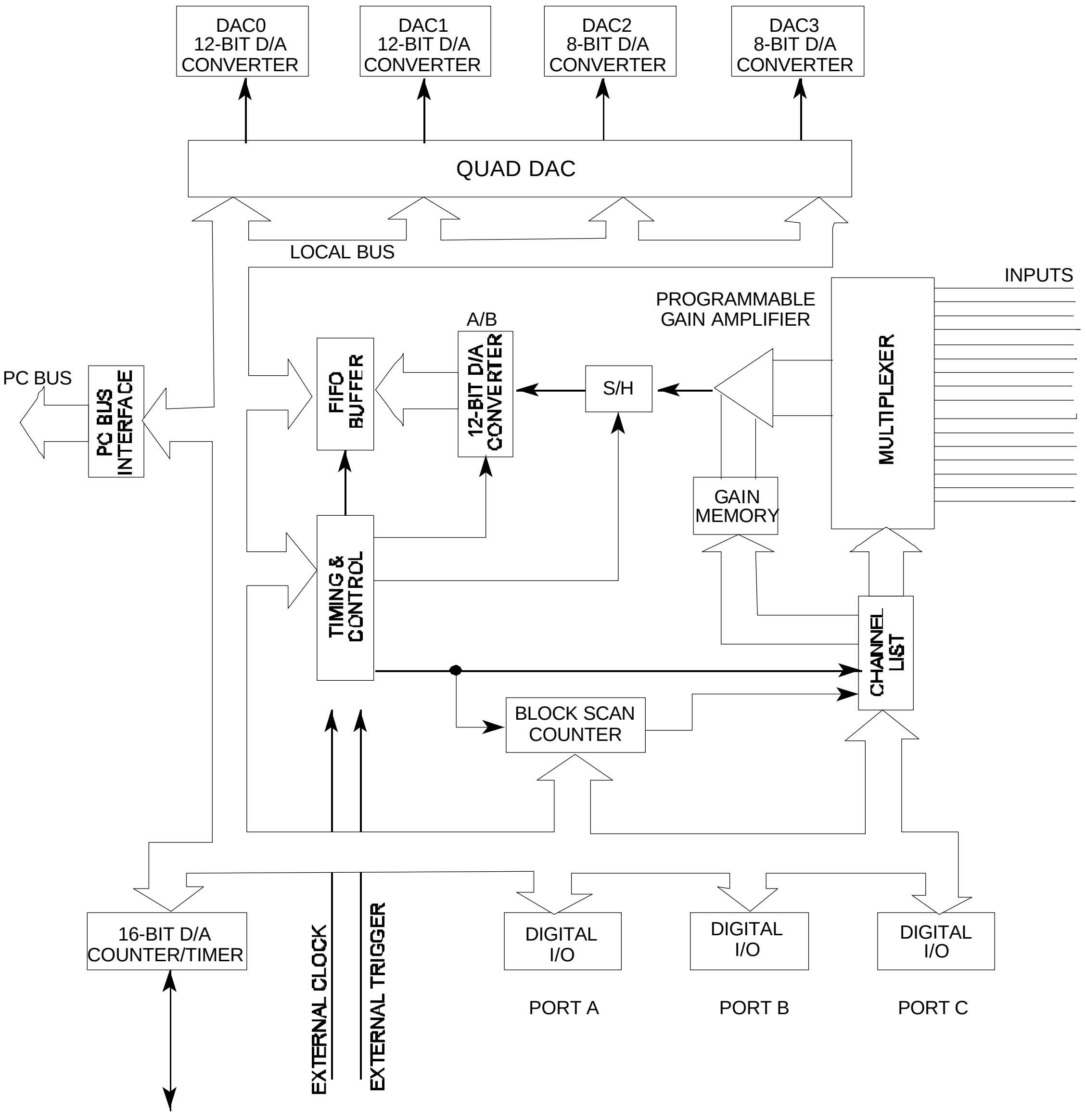


Figure 2 - 1. PC104-30 Board Block Diagram

Bus Interface

The bus interface is responsible for three functions:

a. The decoding of the board's base address, which is set by a DIP switch.

b. The generation of interrupts. Interrupts can be generated under one of three register selectable conditions:

- The end of each A/D conversion.
- The end of a DMA block.
- On each pulse from the uncommitted counter/timer.
- External Trigger (This Feature will only be available on the PC104-30 Rev 1C Boards).
- External Clock (This Feature will only be available on the PC104-30 Rev 1C Boards).

c. The generation of DMA signals. DMA operations are described later in this chapter.

Timing and Control

The timing and control subsection is responsible for the generation of A/D strobes, and also contains an uncommitted counter/timer which can be used for signal generation, or as a frequency or pulse period counter. A/D strobes cause the A/D converter to begin a conversion. A simplified block diagram of this section is shown in

Figure 2 - 2.

A/D strobes may be selected under program control to be either hardware or software strobes.

a. **Software Strobes**. Software strobes are generated by a write operation to a control register. They hence allow a single conversion to be started under program control.

b. **Hardware Strobes**. The source of hardware strobes is register selected from one of two sources:

- The external clock, which is obtained from the J1 connector. This is a TTL level signal. Conversions are started on positive edges of this signal.
- The internal clock. This is derived from a crystal controlled oscillator, which operates at 8MHz. The 8MHz signal is then divided down by a programmable ratio. The internal clock can be set to divide a 2 MHz input from software for compatibility with older boards.

The timing and control section contains four major subsections:

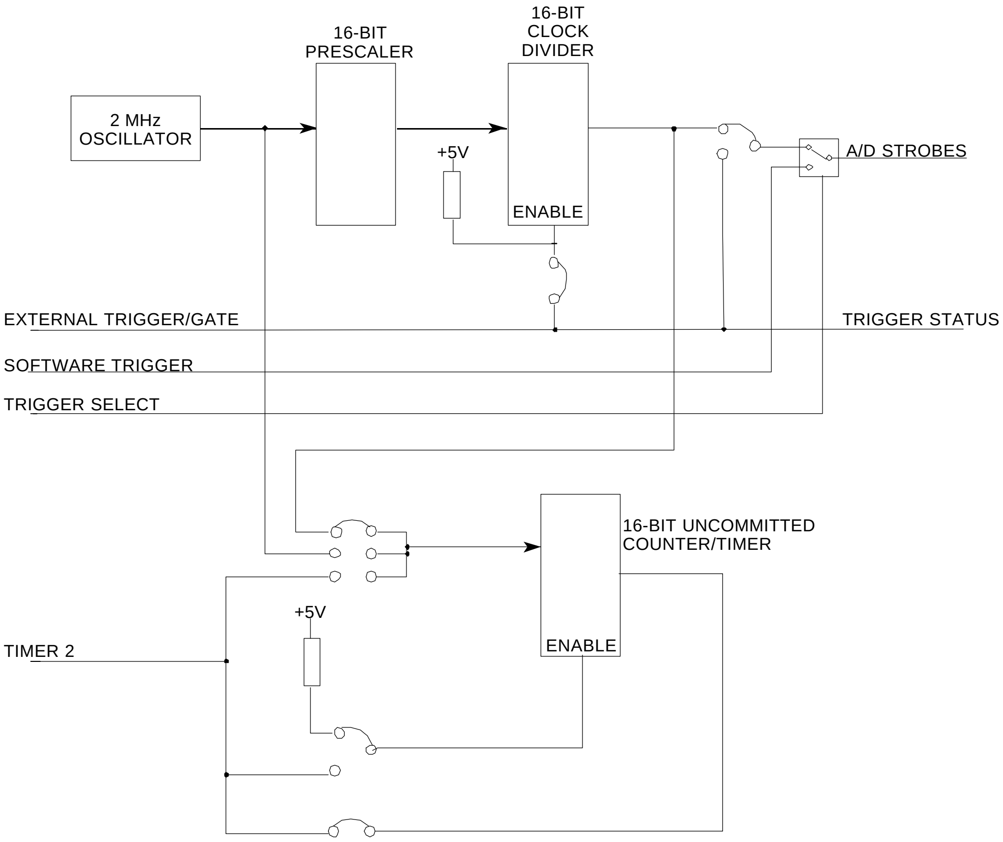


Figure 2 - 2. Timing and Control Block Diagram

Crystal Oscillator

The master clock for the A/D is software selectable to either a 2MHz signal or 8MHz signal, which is derived from an 8MHz temperature controlled crystal oscillator. This is used to drive the clock prescaler. This is a 16 bit counter, the output of which drives the clock divider.

Clock Divider

The clock divider divides the signal from the prescaler by a programmable ratio. No strobes are generated unless the divider is enabled. This allows the start of a set continuous conversion to be synchronised to the trigger input. Note that this divider is active only for an internally generated clock signal.

Clock Selection Multiplexer

The clock selection multiplexer determines whether the A/D strobe signal is derived from the hardware clock (derived either from the internal clock or from the external input) or from the software clock. The software clock is generated by a write operation to a control register and hence allows a single conversion to be started under program control.

Counter/Timer

The PC104-30 contains an uncommitted counter/timer, connected as shown in Figure 2 - 2. This can be configured by software to count either:

a. Pulses from the 2 MHz/8 MHz master oscillator,

b. Pulses from the output of the clock divider, or

c. Pulses from an external source.

This allows the counter to be used for measuring frequency, or for generating a frequency. Counting can also be enabled from an external source, to allow the pulse width to be measured.

A/D Operations

Configuring the Board

Make sure that the configuration registers have been programmed correctly for the input type and range you require. Also ensure that the ADC clock source bits have been programmed.

Sampling Data

A/D operations proceed as follows:

1. The board is initialised as follows:
   a. Setup the IRQ and DMA Request and DACK Lines using the IMUX/DRQ Init Register (Offset 18) and the IGATE/DACK Register (Offset 19).
   b. The appropriate clock and trigger modes are selected, and the clock divider programmed.
   c. The A/D buffer and the trigger system are reset.
   d. The sequence of channels to be converted is written to the channel register.
   e. The required gain settings are written to the gain memory.
2. The system is then enabled, either by a trigger command or by an external signal.
3. As soon as conversions are enabled, A/D conversions start. These conversions occur at the rate set by either the external clock or the internal clock and the value programmed into the clock divider.
4. Conversions continue until the board is disabled. There are two methods of transferring data from the A/D to memory, they are:

Simple Polled I/O

Polled I/O is the easiest method of data transfer, and operates as follows:

1. The program continuously monitors the status register, until data is available.
2. The data from the A/D conversion is then read, and stored in the PC's memory by the program.
3. This process repeats until the required number of samples have been read.

Polled I/O, has the advantage of extreme simplicity, but also has two disadvantages:

- Transfer speed is limited by the speed of the CPU in the host PC.
- While polled I/O is being performed, the CPU is totally dedicated to this process, and cannot deal with anything else (such as keyboard input). Note that for this reason simple polled I/O is generally not suitable for use with multi-tasking operating systems such as Windows ‘95/98/NT4, Linux.

Polled I/O is generally used for single conversions, or continuous conversions at low sampling rates (less than 3 kHz).

Single Block DMA

DMA stands for Direct Memory Access, and operates as follows:

1. The host PC's DMA hardware is first set up with the address of the memory into which the A/D samples are to be put, and the number of samples to be obtained. The board is then initialised, and sampling started, as described above. The program can then continue with any other task, such as checking the keyboard for input.
2. When the A/D buffer contains data, a DMA cycle is initiated.
3. This DMA cycle reads the data from the A/D buffer, and stores it in the PC's memory, without the CPU taking any action.
4. This process repeats until the required number of samples have been read.
5. The program checks the PC104-30 to see if all the required samples have been transferred. Note that this check can be done at any time, unlike for polled I/O, where the A/D status must be checked quickly enough to ensure that data is read before the next A/D conversion completes.

The primary advantage of DMA operation is the very high transfer rate.

The number of samples which can be acquired in a single block is limited by the PC hardware to 65536. The PC104-30 however has special provision for chaining as many blocks as required. This is described in the next section.

Multi Block DMA

The PC104-30 can make use of two DMA channels that are selected by jumpers on the PC104-30 board. Multi block DMA proceeds as follows:

1. The area of memory into which the samples are to be transferred is split into as many blocks of 64k as required. In practice the PC104/AT can transfer up to 128k, but many operating systems provide data in blocks of 64k, as this is the segment size used by the processor in the PC104/AT.
2. The address of the first block is then programmed into the first channel of the DMA controller, the address of the second block is programmed into the second DMA channel, and the A/D sampling initiated. The PC104-30 then begins to transfer A/D samples into memory via the first DMA channel.

3. When the DMA controller reaches the end of the first block, it generates a TC (Terminate Cycle) signal. The PC104-30 uses this to change to the second DMA channel, and to set a status bit indicating block completion. Samples are now transferred via the second DMA channel into the second block of memory. If the board is configured to generate an end of DMA block interrupt, it occurs at the same time as this changeover.
4. Once the data acquisition program detects the end of block condition it programs the address of the next block into the first channel of the DMA controller. When the second channel of DMA reaches the end of its block, DMA swaps back to the first channel. This process continues until all blocks have been filled.

Note that the DMA system must be reprogrammed before the memory block used by the DMA channel becomes full. Under normal circumstances this requires a response time of less than 0.16 s.

Block Mode Triggering

The PC104-30 can operate in one of two modes, ie. normal and block modes.

Normal Mode

In normal mode, for each A/D strobe, one A/D conversion is performed. This is illustrated in Figure 2 - 3. Note that in normal mode, the inputs are sampled prior to each individual A/D conversion.

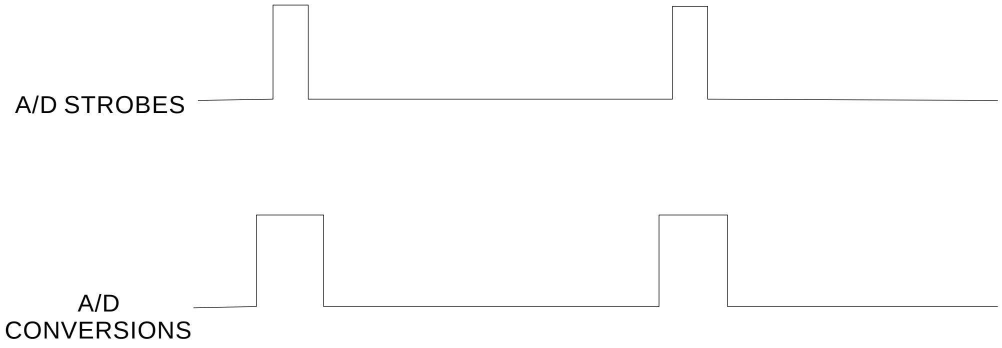


Figure 2 - 3. Normal Mode Operation

Block Trigger Mode

In block trigger mode, for each A/D strobe, as many A/D conversions as are programmed into the block counter are performed. This value may range from 2 to 256. In block mode operation, the inputs are sampled only on each A/D strobe, and NOT prior to each individual A/D conversion.

Block mode operation is not available for PC104-30 input channels operating at a gain of 1000.

Block mode operation is shown in Figure 2 - 4, for a block count of 3.

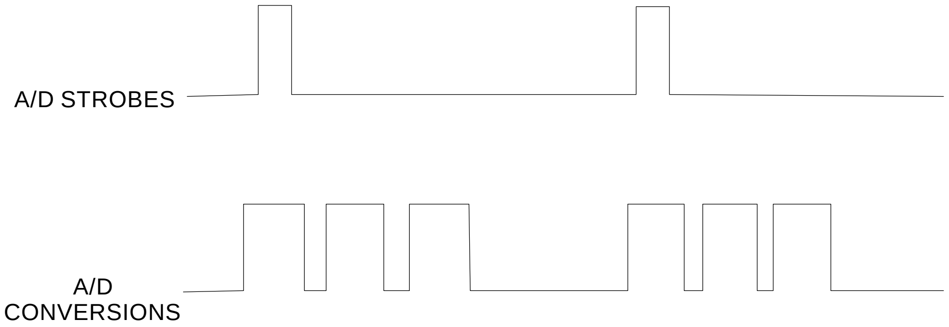


Figure 2 - 4. Block Mode Operation

Operation of the Channel List

The channel list allows the PC104-30 to convert a series of channels, without program intervention. For example, if you want to convert the voltages on channel 2, 15 and 6, in that order, write the following values to the channel register:

2

15

6

When the PC104-30 is triggered, channel 2 will be converted first, then channel 15, then channel 6. The channel list will then loop round, and start at 2 again.

Note that the sequence of channels to be converted may be no longer than 31.

Digital I/O

The digital I/O section of the PC104-30 consists of three 8-bit ports (ports A, B and C), which can be configured in various operating modes. The digital I/O portion of the PC104-30 emulates, and is 100% compatible with, an 8255 type PPI (Programmable Peripheral Interface).

The three ports are divided into two groups, group A (consisting of port A and the upper half of port C) and group B (consisting of port B and the lower half of port C). Each of these groups can be individually configured into one of the following three operating modes: Note that at power up, the interface is set to mode 0.

Mode 0 (Basic Input/Output)

Mode 0 characteristics are:

- Two 8-bit ports (A and B) and two 4-bit ports (upper and lower halves of
- port C).
- Any port can operate either as an input or an output.
- This mode of operation provides simple I/O operations. Any port can be used either for input or output, but not for both.

Mode 1 (Strobed Input/Output)

Mode 1 characteristics are:

- Two groups, each consisting of one 8-bit, and one 4-bit port.
- The 4-bit port is used for control and status of the 8-bit port.
- The 8-bit port may be used for either input or output.
- Input and output operations are latched.
- This mode of operation provides I/O operations with a simple handshake protocol.

Mode 2 (Strobed Bi-directional Input/Output)

Mode 2 characteristics are:

- One 8-bit bi-directional port, and one 5 bit control port.
- Can be used in group A only.
- The 5-bit port is used for control and status of the 8-bit port.
- Input and output operations are latched.
- Port B can still be used in mode 0.
- This mode of operation provides a means for bi-directional I/O operations on a single 8-bit port.

Bit definitions for each mode are described in Chapter 4. Programming of the various modes is described in Chapter 5.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3: Configuring the PC104-30 Board

Introduction

The PC104-30 board can be configured for various user requirements. This configuration is set by software or by the position of the various mini-jumps on the board. Each set of mini-jumps controls a specific function of the board. For software selectable settings, the name of the appropriate EDR function is listed in brackets. These are as follows:

a. **Bus Interface**. The base address of the board, the DMA level and the interrupt level used by the board can be set. The base address is factory set to 700H, the DMA levels 5/6, and the interrupt level to 5. This allows operation in a standard PC/AT which contains only conventional boards (multifunction boards, disk controller boards display boards, etc.). Modification may be required if more specialised boards (other scientific boards, certain backup systems, etc.) are installed. Note that both the DMA request and the interrupt line are electronically disconnected from the PC bus unless specifically enabled by software, even if an interrupt level selection jumper or a DMA level selection jumper is installed.

b. **D/A Operation**. The output range of all four D/A converters may be selected independently by software (use EDR_Set DAOut Range). All D/A outputs default to bipolar output, -10 to + 10V.

c. **A/D Operation**. On the 100K boards with programmable gain (G and GA), the A/D range can be switched between ±5V and 0 - 10V from software (use EDR_Set ADIn Range). The ±10V range must be selected with a jumper. On the 330K programmable gain boards (F and FA), the A/D range can be switched between ±5V and ±10V from software.

   In addition, the gain boards may be configured, by software, to provide either 16 single ended or 8 differential inputs (use EDR_Set ADIn Type.

d. **Uncommitted Counter/Timer.** The PC104-30 contains a single 16-bit counter, which can be used in a variety of roles, including counting events, generating pulses, and measuring frequency. The functions EDR_CT ClockSource, EDR_CT GateSource and EDR_CT Configure, configure the boards counters (timer circuitry).

e. **Output connector**. +5V (200mA max) and digital ground are available on the output connector.

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
0H	on	on	on	on	on	on	on	on
20H	on	on	on	on	on	on	on	off
40H	on	on	on	on	on	on	off	on
60H	on	on	on	on	on	on	off	off
80H	on	on	on	on	on	off	on	on
A0H	on	on	on	on	on	off	on	off
C0H	on	on	on	on	on	off	off	on
E0H	on	on	on	on	on	off	off	off
100H	on	on	on	on	off	on	on	on
120H	on	on	on	on	off	on	on	off
140H	on	on	on	on	off	on	on	off
160H	on	on	on	on	off	on	off	off
180H	on	on	on	on	off	off	on	on
1A0H	on	on	on	on	off	off	on	off
1C0H	on	on	on	on	off	off	off	on
1E0H	on	on	on	on	off	off	off	off
200H	on	on	on	off	on	on	on	on
220H	on	on	on	off	on	on	on	off
240H	on	on	on	off	on	on	off	on
260H	on	on	on	off	on	on	off	off
280H	on	on	on	off	on	off	on	on
2A0H	on	on	on	off	on	off	on	off
2C0H	on	on	on	off	on	off	off	on
2E0H	on	on	on	off	on	off	off	off
300H	on	on	on	off	off	on	on	on

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
320H	on	on	on	off	off	on	on	off
340H	on	on	on	off	off	on	off	on
360H	on	on	on	off	off	on	off	off
380H	on	on	on	off	off	off	on	on
3A0H	on	on	on	off	off	off	on	off
3C0H	on	on	on	off	off	off	off	on
3E0H	on	on	on	off	off	off	off	off
400H	on	on	off	on	on	on	on	on
420H	on	on	off	on	on	on	on	off
440H	on	on	off	on	on	on	off	on
460H	on	on	off	on	on	on	off	off
480H	on	on	off	on	on	off	on	on
4A0H	on	on	off	on	on	off	on	off
4C0H	on	on	off	on	on	off	off	on
4E0H	on	on	off	on	on	off	off	off
500H	on	on	off	on	off	on	on	on
520H	on	on	off	on	off	on	on	off
540H	on	on	off	on	off	on	off	on
560H	on	on	off	on	off	on	off	off
580H	on	on	off	on	off	off	on	on
5A0H	on	on	off	on	off	off	on	off
5C0H	on	on	off	on	off	off	off	on
5E0H	on	on	off	on	off	off	off	off
600H	on	on	off	off	on	on	on	on
620H	on	on	off	off	on	on	on	off
640H	on	on	off	off	on	on	off	on
660H	on	on	off	off	on	on	off	off
680H	on	on	off	off	on	off	on	on
6A0H	on	on	off	off	on	off	on	off
6C0H	on	on	off	off	on	off	off	on
6E0H	on	on	off	off	on	off	off	off
700H	on	on	off	off	off	on	on	on
720H	on	on	off	off	off	on	on	off

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
740H	on	on	off	off	off	on	off	on
760H	on	on	off	off	off	on	off	off
780H	on	on	off	off	off	off	on	on
7A0H	on	on	off	off	off	off	on	off
7C0H	on	on	off	off	off	off	off	on
7E0H	on	on	off	off	off	off	off	off
800H	on	off	on	on	on	on	on	on
820H	on	off	on	on	on	on	on	off
840H	on	off	on	on	on	on	off	on
860H	on	off	on	on	on	on	off	off
880H	on	off	on	on	on	off	on	on
8A0H	on	off	on	on	on	off	on	off
8C0H	on	off	on	on	on	off	off	on
8E0H	on	off	on	on	on	off	off	off
900H	on	off	on	on	off	on	on	on
920H	on	off	on	on	off	on	on	off
940H	on	off	on	on	off	on	off	on
960H	on	off	on	on	off	on	off	off
980H	on	off	on	on	off	off	on	on
9A0H	on	off	on	on	off	off	on	off
9C0H	on	off	on	on	off	off	off	on
9E0H	on	off	on	on	off	off	off	off
A00H	on	off	on	off	on	on	on	on
A20H	on	off	on	off	on	on	on	off
A40H	on	off	on	off	on	on	off	on
A60H	on	off	on	off	on	on	off	off
A80H	on	off	on	off	on	off	on	on
AA0H	on	off	on	off	on	off	on	off
AC0H	on	off	on	off	on	off	off	on
AE0H	on	off	on	off	on	off	off	off
B00H	on	off	on	off	off	on	on	on
B20H	on	off	on	off	off	on	on	off
B40H	on	off	on	off	off	on	off	on

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
B60H	on	off	on	off	off	on	off	off
B80H	on	off	on	off	off	off	on	on
BA0H	on	off	on	off	off	off	on	off
BC0H	on	off	on	off	off	off	off	on
BE0H	on	off	on	off	off	off	off	off
C00H	on	off	off	on	on	on	on	on
C20H	on	off	off	on	on	on	on	off
C40H	on	off	off	on	on	on	off	on
C60H	on	off	off	on	on	on	off	off
C80H	on	off	off	on	on	off	on	on
CA0H	on	off	off	on	on	off	on	off
CC0H	on	off	off	on	on	off	off	on
CE0H	on	off	off	on	on	off	off	off
D00H	on	off	off	on	off	on	on	on
D20H	on	off	off	on	off	on	on	off
D40H	on	off	off	on	off	on	off	on
D60H	on	off	off	on	off	on	off	off
D80H	on	off	off	on	off	off	on	on
DA0H	on	off	off	on	off	off	on	off
DC0H	on	off	off	on	off	off	off	on
DE0H	on	off	off	on	off	off	off	off
E00H	on	off	off	off	on	on	on	on
E20H	on	off	off	off	on	on	on	off
E40H	on	off	off	off	on	on	off	on
E60H	on	off	off	off	on	on	off	off
E80H	on	off	off	off	on	off	on	on
EA0H	on	off	off	off	on	off	on	off
EC0H	on	off	off	off	on	off	off	on
EE0H	on	off	off	off	on	off	off	off
F00H	on	off	off	off	off	on	on	on
F20H	on	off	off	off	off	on	on	off
F40H	on	off	off	off	off	on	off	on
F60H	on	off	off	off	off	on	off	off

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
F80H	on	off	off	off	off	off	on	on
FA0H	on	off	off	off	off	off	on	off
FC0H	on	off	off	off	off	off	off	on
FE0H	on	off	off	off	off	off	off	off
1000H	off	on	on	on	on	on	on	on
1020H	off	on	on	on	on	on	on	off
1040H	off	on	on	on	on	on	off	on
1060H	off	on	on	on	on	on	off	off
1080H	off	on	on	on	on	off	on	on
10A0H	off	on	on	on	on	off	on	off
10C0H	off	on	on	on	on	off	off	on
10E0H	off	on	on	on	on	off	off	off
1100H	off	on	on	on	off	on	on	on
1120H	off	on	on	on	off	on	on	off
1140H	off	on	on	on	off	on	off	on
1160H	off	on	on	on	off	on	off	off
1180H	off	on	on	on	off	off	on	on
11A0H	off	on	on	on	off	off	on	off
11C0H	off	on	on	on	off	off	off	on
11E0H	off	on	on	on	off	off	off	off
1200H	off	on	on	off	on	on	on	on
1220H	off	on	on	off	on	on	on	off
1240H	off	on	on	off	on	on	off	on
1260H	off	on	on	off	on	on	off	off
1280H	off	on	on	off	on	off	on	on
12A0H	off	on	on	off	on	off	on	off
12C0H	off	on	on	off	on	off	off	on
12E0H	off	on	on	off	on	off	off	off
1300H	off	on	on	off	off	on	on	on
1320H	off	on	on	off	off	on	on	off
1340H	off	on	on	off	off	on	off	on
1360H	off	on	on	off	off	on	off	off
1380H	off	on	on	off	off	off	on	on

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
13A0H	off	on	on	off	off	off	on	off
13C0H	off	on	on	off	off	off	off	on
13E0H	off	on	on	off	off	off	off	off
1400H	off	on	off	on	on	on	on	on
1420H	off	on	off	on	on	on	on	off
1440H	off	on	off	on	on	on	off	on
1460H	off	on	off	on	on	on	off	off
1480H	off	on	off	on	on	off	on	on
14A0H	off	on	off	on	on	off	on	off
14C0H	off	on	off	on	on	off	off	on
14E0H	off	on	off	on	on	off	off	off
1500H	off	on	off	on	off	on	on	on
1520H	off	on	off	on	off	on	on	off
1540H	off	on	off	on	off	on	off	on
1560H	off	on	off	on	off	on	off	off
1580H	off	on	off	on	off	off	on	on
15A0H	off	on	off	on	off	off	on	off
15C0H	off	on	off	on	off	off	off	on
15E0H	off	on	off	on	off	off	off	off
1600H	off	on	off	off	on	on	on	on
1620H	off	on	off	off	on	on	on	off
1640H	off	on	off	off	on	on	off	on
1660H	off	on	off	off	on	on	off	off
1680H	off	on	off	off	on	off	on	on
16A0H	off	on	off	off	on	off	on	off
16C0H	off	on	off	off	on	off	off	on
16E0H	off	on	off	off	on	off	off	off
1700H	off	on	off	off	off	on	on	on
1720H	off	on	off	off	off	on	on	off
1740H	off	on	off	off	off	on	off	on
1760H	off	on	off	off	off	on	off	off
1780H	off	on	off	off	off	off	on	on
17A0H	off	on	off	off	off	off	on	off

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
17C0H	off	on	off	off	off	off	off	on
17E0H	off	on	off	off	off	off	off	off
1800H	off	off	on	on	on	on	on	on
1820H	off	off	on	on	on	on	on	off
1840H	off	off	on	on	on	on	off	on
1860H	off	off	on	on	on	on	off	off
1880H	off	off	on	on	on	off	on	on
18A0H	off	off	on	on	on	off	on	off
18C0H	off	off	on	on	on	off	off	on
18E0H	off	off	on	on	on	off	off	off
1900H	off	off	on	on	off	on	on	on
1920H	off	off	on	on	off	on	on	off
1940H	off	off	on	on	off	on	off	on
1960H	off	off	on	on	off	on	off	off
1980H	off	off	on	on	off	off	on	on
19A0H	off	off	on	on	off	off	on	off
19C0H	off	off	on	on	off	off	off	on
19E0H	off	off	on	on	off	off	off	off
1A00H	off	off	on	off	on	on	on	on
1A20H	off	off	on	off	on	on	on	off
1A40H	off	off	on	off	on	on	off	on
1A60H	off	off	on	off	on	on	off	off
1A80H	off	off	on	off	on	off	on	on
1AA0H	off	off	on	off	on	off	on	off
1AC0H	off	off	on	off	on	off	off	on
1AE0H	off	off	on	off	on	off	off	off
1B00H	off	off	on	off	off	on	on	on
1B20H	off	off	on	off	off	on	on	off
1B40H	off	off	on	off	off	on	off	on
1B60H	off	off	on	off	off	on	off	off
1B80H	off	off	on	off	off	off	on	on
1BA0H	off	off	on	off	off	off	on	off
1BC0H	off	off	on	off	off	off	off	on

Table 3 - 1: DIP Switch Setting

Base Address	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
1BE0H	off	off	on	off	off	off	off	off
1C00H	off	off	off	on	on	on	on	on
1C20H	off	off	off	on	on	on	on	off
1C40H	off	off	off	on	on	on	off	on
1C60H	off	off	off	on	on	on	off	off
1C80H	off	off	off	on	on	off	on	on
1CA0H	off	off	off	on	on	off	on	off
1CC0H	off	off	off	on	on	off	off	on
1CE0H	off	off	off	on	on	off	off	off
1D00H	off	off	off	on	off	on	on	on
1D20H	off	off	off	on	off	on	on	off
1D40H	off	off	off	on	off	on	off	on
1D60H	off	off	off	on	off	on	off	off
1D80H	off	off	off	on	off	off	on	on
1DA0H	off	off	off	on	off	off	on	off
1DC0H	off	off	off	on	off	off	off	on
1DE0H	off	off	off	on	off	off	off	off
1E00H	off	off	off	off	on	on	on	on
1E20H	off	off	off	off	on	on	on	off
1E40H	off	off	off	off	on	on	off	on
1E60H	off	off	off	off	on	on	off	off
1E80H	off	off	off	off	on	off	on	on
1EA0H	off	off	off	off	on	off	on	off
1EC0H	off	off	off	off	on	off	off	on
1EE0H	off	off	off	off	on	off	off	off
1F00H	off	off	off	off	off	on	on	on
1F20H	off	off	off	off	off	on	on	off
1F40H	off	off	off	off	off	on	off	on
1F60H	off	off	off	off	off	on	off	off
1F80H	off	off	off	off	off	off	on	on
1FA0H	off	off	off	off	off	off	on	off
1FC0H	off	off	off	off	off	off	off	on
1FE0H	off	off	off	off	off	off	off	off

Changing the Configuration

The DIP switch and jumpers may be located either from the component layout in Appendix C, or from the labels on the PC104-30 board itself.

To change the jumper or DIP switch settings, proceed as follows:

1. Switch off the computer.
2. Remove the board.
3. Change the required jumpers or DIP switch settings.
4. Reinsert the board in the PC.
5. Power up, and run a program (such as Waveview), which executes the PC104-30 diagnostics routines.

Bus Interface Configuration

Base Address

The base address setting is controlled by the DIP switch on the board. The address is factory set to 700H. The board occupies 32 consecutive locations. Table 3-1 shows the DIP switch settings for each base address.

Interrupt Level

The interrupt level is selected under software control using the IMUX (Offset 18) and IGATE (Offset 19) Registers.

Selection of Interrupt Level

In a standard PC, interrupts are allocated as follows:

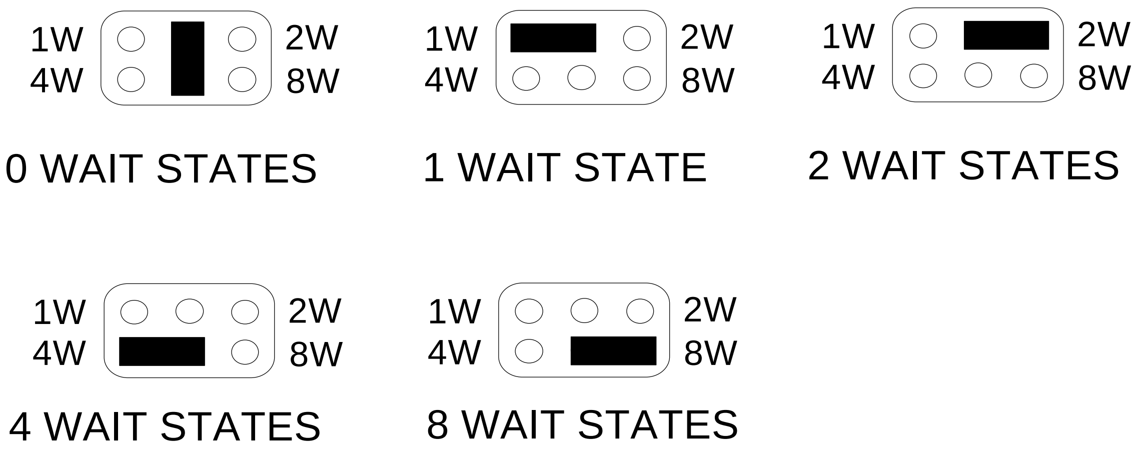
- level 3: Used by COM2 (if installed)
- level 4: Used by COM1 (if installed)
- level 5: Used by fixed disks (XT and AT)
- level 7: Used by LPT1 (if installed)
- level 10: PC2 Mouse if installed
- level 11: Unused
- level 12: Unused
- level 14: Used by fixed disks (Primary)
- level 15: Used by fixed disks (Secondary) if installed

An interrupt level which is not already used must be selected. No interrupts are selected on Power Up. Note that unless the interrupts are specifically enabled by software, the interrupt output from the board is tri-stated (does not have any effect on the PC104 bus).

Wait State Jumper

Some PC104 Systems have very high I/O bus cycles not compliant to the PC104 Standard. In this case it is necessary to slow down these cycles when the computer accesses the PC104-30F/G Board. Additional wait states can be set by means of a jumper on the PC104-30 Board.

Additional wait states can be inserted in the I/O bus cycle by changing the jumper setting on JP2 on the PC104-30. This jumper is marked 'Wait State Jumper' on the PC104-30 board. Refer to the figure below for the wait state jumper settings. Note that the factory default setting is zero wait states.



0 WAIT STATES

1 WAIT STATE

2 WAIT STATES

4 WAIT STATES

8 WAIT STATES

Figure 3 - 1. PC104-30 Wait State Jumper

Interrupt Source

The source of PC104-30 interrupts can be software configured to be one of five events:

a. **End of Conversion**. In this case, an interrupt is generated each time that the PC104-30 completes an A/D conversion.

b. **End of DMA block**. In this case, an interrupt is generated on completion of each DMA block. This can be used to signal the end of an operation or to signal that the DMA controller must be reprogrammed in dual DMA channel "ping-pong" operations.

> End of DMA block interrupts are only generated if the PC104-30 registers are programmed correctly. See the description of the DMA mode bit for more information.

c. **On each pulse from the counter/timer**. If this jumper option is selected, an interrupt is generated on each pulse from the uncommitted counter/timer. The counter/timer is usually configured to generate a constant frequency, as discussed in Chapter 5.

d. **External Trigger**: Interrupts will be generated on a positive edge of the External Trigger Line. This Option is only available on the PC104-30F/G Rev 1C Boards.

e. **External Clock**: Interrupts will be generated on a positive edge of the External Clock Line. This Option is only available on the PC104-30F/G Rev 1C Boards.

The required setting can be selected by running the hardware configuration program supplied, or by using EDR_SetInterruptSource in your own programs.

DMA Level

On the PC104-30, the DMA setting is configured via Software. Upon Power-up the all the DMA Channels are Tri-stated.

Note that the default setting is for dual channel mode on channels 5 and 6.

CAUTION

Do not configure both the Primary and Secondary DMA Channel to the same DMA Channel. If Single DMA is used only configure the Primary Dma Channel and disable the secondary DMA Channel.

Selection of DMA Levels

In a standard PC, DMA channels are allocated as follows:

• level 1:	Unused
• level 3:	Used by fixed disks (XT and AT)
• level 5:	Unused
• level 6:	Unused
• level 7:	Unused

A DMA level which is not already used must be selected. Note that unless DMA is specifically enabled by software, the DMA outputs from the board are tri-stated (do not have any effect on the PC bus).

D/A Selections

The four D/A converters may be software configured for either monopolar (0 to 10V and 0 to 20V) or bipolar (-10 to +10V and -5 to +5V) output ranges. All four DAC channel output ranges can be configured independently by software. Use EDR_Set DAOut Range to do this in your programs.

A/D Configuration

The following three aspects of A/D operation can be configured:

- A/D input mode (single ended or differential)
- A/D voltage range

- A/D clock/trigger

A/D Input Mode

The PC104-30F, FA, G and GA models can provide either 16 single ended or eight differential inputs. The use of differential inputs is recommended in environments with high levels of electrical noise, when using long lines to connect the analogue inputs, or for any input operating at a gain of greater than 10. Default power-up configuration is for single ended inputs. The required mode is selected by software using EDR_SetADInType.

A/D Voltage Range Setting

On the G and GA models, EDR_SetADInRange can switch between ±5V and 0V to 10V. LK1 must be used to achieve the ±10V range (see below for jumper settings for the **G and GA models** only). On the F and FA models, the range can be switched between ±5V and ±10V from software.

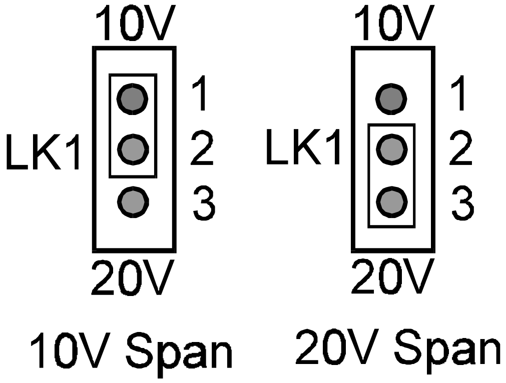


Figure 3 - 2. PC104-30G/GA Span Jumper

On the PC104-30 F/G board, you can refer to the text written on the silk-screen layer with regard to jumper settings.

CAUTION !

CHANGING JUMPER LK1 TO THE 20 V POSITION DOES NOT ENABLE YOU TO SELECT UNIPOLAR RANGE 0 TO 10 V.

Note that the default range setting is -5 to +5V.

A/D Clock/Trigger

Figure 3 - 3 shows a partial schematic of the PC104-30 clock/trigger system.

Counter 0 of the 8254 serves as the A/D clock prescaler, and counter 1, as the A/D clock divider. Counter 2 is the uncommitted counter/timer. Software (use EDR_Set ADCClock Trigger) is used to configure the source of A/D clock pulses. Four possible configurations are supported:

a. **Internal Clock**. A/D clock pulses are obtained from the clock divider. The external trigger input has no direct effect on the operation of the board, but can be read by software. The A/D clock frequency is the frequency of the 8 MHz master clock divided by the values programmed into the A/D prescaler and the A/D clock divider. The EDR constant for this mode is EDR_ADC_INTERNAL.

b. **Internal Clock/External Trigger**. A/D clock pulses are obtained from the clock divider, and the external trigger input is a TTL compatible clock enable. If the external trigger input is logically high, A/D clock pulses are produced. If the external trigger input is logically low, then clock pulses are not produced. The A/D clock frequency is the frequency of the 8 MHz master clock divided by the values programmed into the A/D prescaler and the A/D clock divider. The EDR constant for this mode is EDR_ADC_EXTTRIG.

c. **External Clock**. A/D clock pulses are obtained from the external trigger input. In this case, the external trigger input serves directly as the source of A/D clock pulses. Note that in this case neither the A/D prescaler or clock divider have any effect on the sample rate. The EDR constant for this mode is EDR_ADC_SLAVE.

d. **Internal Clock to ADC and External Trigger**. This mode is similar to (a) but the clock pulses are also outputted on the external trigger line. This can be used to clock other boards in mode (c). The EDR constant for this mode is EDR_ADC_MASTER. This is the default configuration.

All the above-mentioned modes can be selected from software. Note that the default setting is for A/D clock pulses obtained from the clock divider and outputted on the external trigger line.

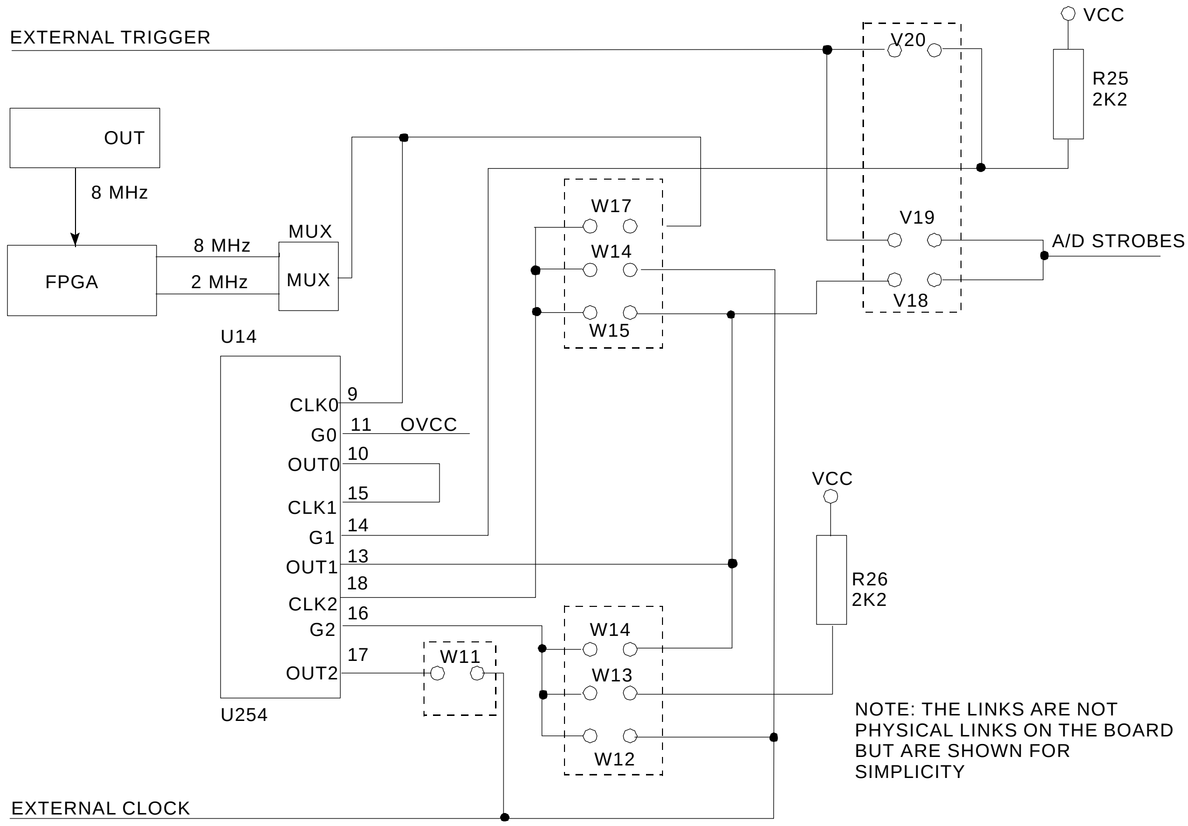


Figure 3 - 3. PC104-30 Clock Generation Schematic

Uncommitted Counter/Timer

The PC104-30 contains an uncommitted counter/timer, which can be used for a variety of tasks, including counting pulses, generating pulses, generating a constant frequency, and measuring frequency and period.

Three aspects of the counter timer operation can be configured:

External Clock Line

The external clock lines can either be an input to, or output from, the counter/timer. This is configured by software. The default configuration is as an output. EDR_CTClockSource configures it as an output for all constants except EDR_CS_EXT. EDR_CTGateSource does not change its configuration unless using EDR_CT_EXTGATE, in which case it is set as an input.

Counter/Timer Clock Source

The counter/timer can be configured using EDR_CT_ClockSource to count pulses from four sources:

a. **Clock source is grounded.** This can be used to make sure the counter does not count. The EDR constant is EDR_CS_GND.

b. **Pulses from the master clock**. In this case, the counter/timer counts pulses from the 2 MHz or 8 MHz master clock oscillator. This is generally used either to generate a frequency, or to measure period. The EDR constants are EDR_CS_2MHz and EDR_CS_8MHz. Remember that the prescaler (counter 0) shares this input. Selecting EDR_CS_2MHz or EDR_CS_8MHz will change the input to the prescaler as well.

c. **Pulses from the clock divider**. In this case, the counter timer counts at the rate set by the clock prescaler and clock divider. This clock is also derived from the master clock oscillator, but can be of a much lower frequency. The EDR constant is EDR_CS_DIVIDER.

d. **Pulses from the external clock line**. To use this option, the eternal clock line must be configured as an input, as described above. This option is generally used to count external events or frequency. The EDR constant is EDR_CS_EXT.

The counter timer clock source is set by software and the default is for counting the master clock.

Counter/Timer Enable

The counter/timer counts only if the gate input is at logical high. The input to this gate can be obtained from three different sources configurable from software using EDR_CTGateSource.

a. **Constantly enabled**. The gate input to the counter timer can be constantly enabled. This is generally used for generating frequencies and pulses, or for counting events. The EDR constant is EDR_CT_ENABLED.

b. **Enabled from the output of the A/D clock divider**. This mode is generally used for measuring frequency. The A/D divider is programmed to generate a pulse of a known length, and the counter/timer set to count input pulses. The count gives a measure of input frequency. The EDR constant is EDR_CT_DIVOUTPUT.

c. **Enabled from the external clock line** . In this case the external clock must be configured as an input, as described above. This configuration is generally used to measure period, in which case the counter is configured to count master clock pulses. The EDR constant is EDR_CT_EXTGATE.

The counter timer enable source is set by software and the default setting is for constant enable.

Chapter 4: Interconnections

Introduction

The PC104-30 plugs into any 16-bit PC104 expansion slot, at connector P1. All boards in the family connect to the user's circuitry at connector J1. This chapter describes these two connectors.

Connections to the PC104 Connectors

PC104-30 boards plug into any 16 Bit PC104 Slot at J1 and J3. All communication to and from the host processor is carried out via this connector.

Analog User Connector [AD-CON]

The PC104-30 is connected to the user interface via an IDC 26 Way male header for the Analog. This connector accommodates the following signals:

- 16 single ended or 8 differential lines (for the F, FA, G and GA models) of analogue input.
- 4 lines of analogue output with remote sense.
- Two Analog Ground Pins for the Analog Inputs Signals and the DAC Output Signals

Figure 4 - 1 shows these connections, together with their pin assignments. Note that the pin connections are shown as seen looking into the connector. Pin 1 is CH1 in the figure below. Do not make use of any numbers on the PC104-30 board.

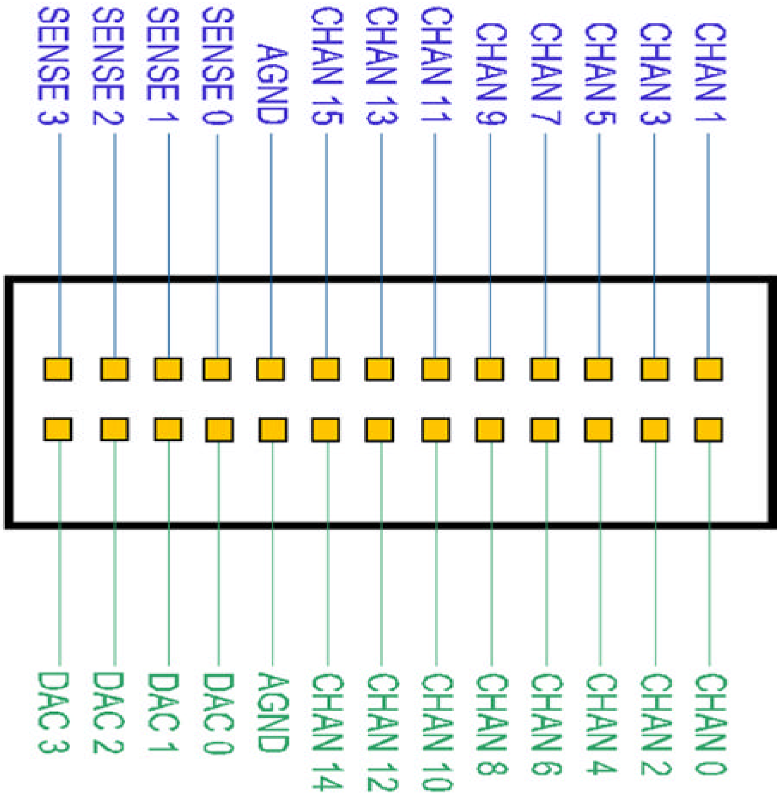


Figure 4 - 1. PC104-30 Analog Connector

Signal Definitions

a. **CH0 - CH15**. These are the analogue input lines. Note that no more than ±10V must be applied to these pins although these lines are protected up to a ±35V in an ON state and ±25V in an OFF state. In differential mode, channel 8 serves as the return line for input channel 0, channel 9 as that for input channel 1 etc.

b. **ANALOGUE GROUND**. Two analogue ground lines are provided. One for the A/D Channels and the other for the D/A Channels. If you are not using the D/A channels, you can use this ground for the A/D Inputs as well. The analogue input lines are measured relative to AGND.

c. **DAC0 OUTPUT**. This is the analogue output line for DAC0.

d. **DAC1 OUTPUT**. This is the analogue output line for DAC1.

e. **DAC2 OUTPUT**. This is the analogue output line for DAC2.

f. **DAC3 OUTPUT**. This is the analogue output line for DAC3.

g. **SENSE0 INPUT**. This line is the Remote Sense Line for DAC0. This line is used if the cable from the PC104-30GA/FA to Load is long. The Sense Line compensates for the voltage drop occurring from the DAC Output to the Load. Connect the DAC0 output to the SENSE0 at the Load. If the Remote Sense Input Line is not used then you MUST short the SENSE0 Line to the DAC0 output line. Failure to do this will result in the DAC output voltage being permanently set to about ±13V.

h. **SENSE1 INPUT**. This line is the Remote Sense Line for DAC1. This line is used if the cable from the PC104-30GA/FA to Load is long. The Sense Line compensates for the voltage drop occurring from the DAC Output to the Load. Connect the DAC1 output to the SENSE1 at the Load. If the Remote Sense Input Line is not used then you MUST short the SENSE1 Line to the DAC1 output line. Failure to do this will result in the DAC output voltage being permanently set to about ±13V.

i. **SENSE2 INPUT**. This line is the Remote Sense Line for DAC2. This line is used if the cable from the PC104-30GA/FA to Load is long. The Sense Line compensates for the voltage drop occurring from the DAC Output to the Load. Connect the DAC2 output to the SENSE2 at the Load. If the Remote Sense Input Line is not used then you MUST short the SENSE2 Line to the DAC2 output line. Failure to do this will result in the DAC output voltage being permanently set to about ±13V.

j. **SENSE3 INPUT**. This line is the Remote Sense Line for DAC3. This line is used if the cable from the PC104-30GA/FA to Load is long. The Sense Line compensates for the voltage drop occurring from the DAC Output to the Load. Connect the DAC3 output to the SENSE3 at the Load. If the Remote Sense Input Line is not used then you MUST short the SENSE3 Line to the DAC3 output line. Failure to do this will result in the DAC output voltage being permanently set to about ±13V.

Digital User Connector [P1 - DIG40]

The PC104-30 Digital Signals are available via an IDC40 Male. Note that the pin connections are shown as seen looking into the connector. Pin 1 is PA1 in the figure below. Do not make use of any numbers on the PC104-30 board.

This connector accommodates the following signals:

- 24 digital I/O lines.
- External Trigger (For Triggering A/D conversions)
- External Clock
- Clock Input 0
- Gate 1 and Out1
- User Counter Timer CLK2
- User Counter Timer GATE2
- User Counter Timer OUT2
- +5V (200mA Max Current)
- Digital Ground

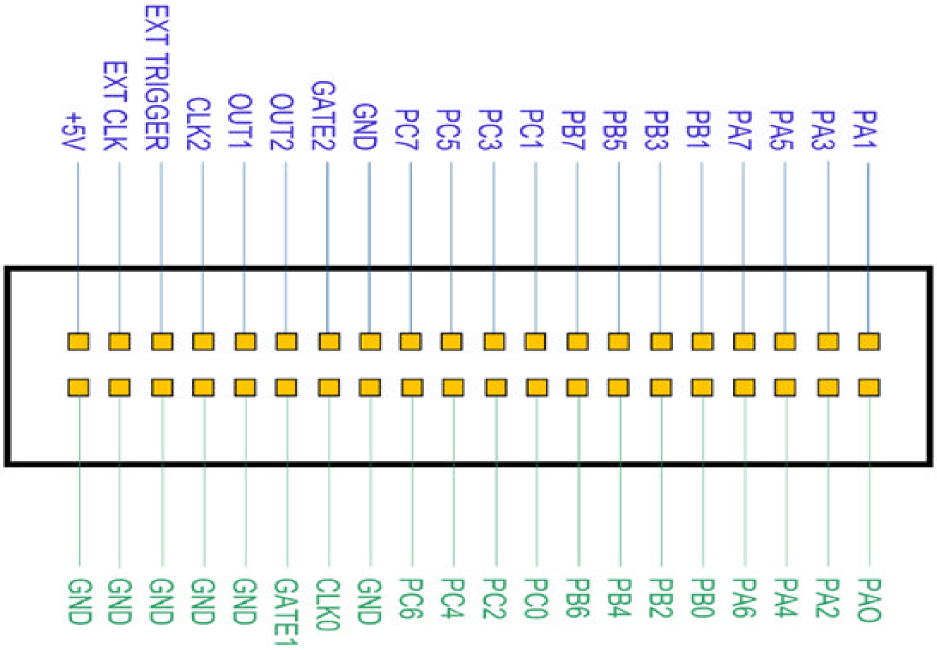


Figure 4 - 2. PC104-30 Digital Connector

Signal Definitions

a. **Port A0 - A7**. This is the first digital I/O port, digital I/O port 0. It is configurable into a number of operating modes under software control.

b. **Port B0 - B7**. Digital I/O port 1. It is configurable into a number of operating modes under software control.

c. **Port C0 - C7**. Digital I/O port 2. It is configurable into a number of operating modes under software control.

d. **External Trigger**. This line is is software selectable to provide a clock or trigger signal to the A/D. It may also be connected to the directly to the IRQ Sub-system to generate Interrupts. It can also be read under software control. It is TTL compatible. This line can also be configured as an output to synchronise boards in master/slave modes.

e. **External Clock**. This line is also software selectable. It can be interfaced to the uncommitted counter/timer, and can be configured to perform a variety of functions, as described in the previous chapter. It can also be connected to the directly to the IRQ Sub-system to generate Interrupts. It may be configured either as an input or output and is also TTL compatible.

f. **CLK0:** Input to Counter Timer CLK0. Used as an input to the Prescaler (Counter 0). Note that this line should not be used if hardware A/D Strobes are used.

g. **GATE1:** This line is used to control the Strobes to the A/D. GATE1 is normally enabled on power-up and is TTL compatible.

h. **OUT1:** This line is used to strobe the A/D and is the output of the DIVIDER Counter Timer (Timer1). Do NOT use this line if Hardware A/D Strobes are used.

i. **CLK2:** This line is the input CLK Line to the User Counter Timer (ie: Counter Timer2). It can be used to count events, measure period/frequency, etc. Note that this line is TTL Compatible.

j. **GATE2:** Gate Control Line of the User Counter Timer (ie: Counter Timer2). It can be used enable/disable the counting on the User Counter Timer.

k. **OUT2:** Output Line of the User Counter Timer (ie: Counter Timer2). It can be used to generate a pulse on terminal count, output constant frequency pulses, etc.

l. **+5V.** This line provides a +5V power supply to the user's interface. Maximum permissible current draw is 250 mA. This line is equipped with a Polyswitch that will open circuit if the max permissible current of 200mA is exceeded.

m. **Digital Ground**. These lines provide the digital ground connection to all the digital signals. Digital ground is the ground return line for the digital inputs and outputs. Any digital circuitry tied to the digital lines should be referenced to these lines. It is internally connected to analogue ground.

Analogue I/O

Recommended Analogue Input Schemes

Analogue signals are input into the PC104-30 either as single ended inputs or differential inputs (F, FA, G and GA models only).

CAUTION

OVERLOADING ANY ANALOGUE INPUT BY MORE THAN 10% MAY CAUSE OTHER INPUT CHANNELS TO BECOME INACCURATE OR NOISY. FOR PC104-30F/G INPUTS OPERATING AT MAXIMUM GAIN, THIS CORRESPONDS TO AN INPUT VOLTAGE OF 5.5 mV.

Single Ended Inputs

In single ended connections (see Figure 4 - 3), input signals share a common low side, which is analogue ground.

This has the advantage of giving the maximum number of inputs. Its major disadvantage is the loss of common mode rejection obtainable from differential mode. Single ended inputs are very sensitive to noise, and should not be used with lead lengths of greater than 18 inches, or for inputs with a gain of greater than 10.

Differential Inputs

In differential input mode (see Figure 4 - 4), two multiplexer switches per channel are used, and the A/D converter measures the difference between the high and low input lines of each channel.

In differential mode, channels 0 and 8 form the high and low inputs of input channel 0, channels 1 and 9 that of input channel 1 etc.

Analogue inputs are limited to a voltage of between -10 and +10V.

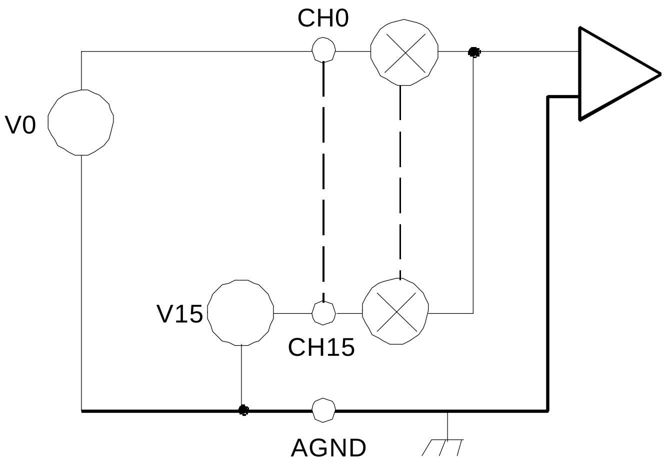


Figure 4 - 3. Single Ended Analogue Inputs

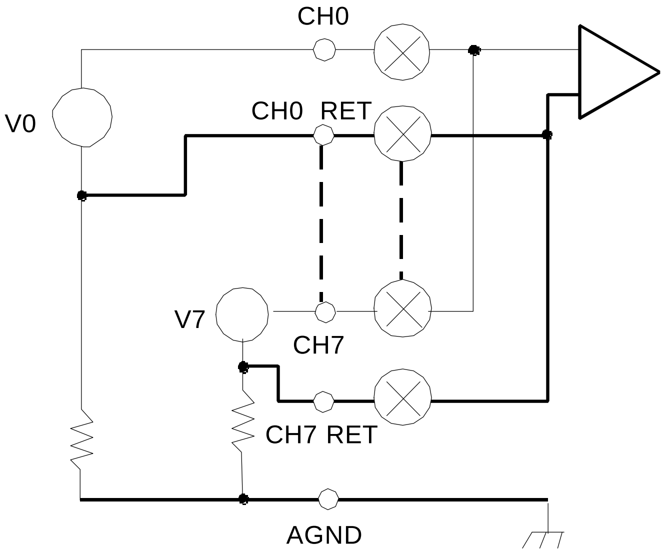


Figure 4 - 4. Differential Analogue Inputs

In differential mode, all signal inputs to the PC104-30 must be referred to ground. This can be done by connecting a 1 to 10 kΩ resistor from the low end of each input to ground.

Analogue Output

The analogue output lines are referenced to the analogue ground line and is software selectable to give either monopolar or bipolar outputs.

Connection Guidelines

The PC104-30 is a very high performance I/O subsystem, and was designed to have low input noise. Its performance may however be severely affected by incorrect connection techniques. This is especially true of noise levels.

Shielded Input Lines

Wherever possible, leads should be shielded. Optimally, each input line should be individually shielded. The shield should be tied to analogue ground at the instrument end of the connection only.

Grounding

If user circuitry is connected to the PC104-30 it is critical to keep the digital and analogue ground separate.

Input Voltages

To maintain the specified accuracy, all inputs to the PC104-30 must be within 110% of full scale.

Source Impedance

To maintain the specified accuracy, all devices connected to the analogue inputs of the PC104-30 must have a source impedance of less than l kΩ.

Digital I/O

The Digital I/O section of the PC104-30 consists of three 8-bit ports (port A, B and C), which can be configured in a variety of operating modes. The digital I/O portion of the PC104-30 emulates, and is fully compatible with, an 8255 type PPI (Programmable Peripheral Interface).

The three ports are divided into two groups, group A (consisting of port A and the upper half of port C) and group B (consisting of port B and the lower half of port C). Each group can be individually configured into one of the three following operating modes:

Take note that at power up, the interface is set to mode 0. For each mode, various functions are assigned to the assorted I/O lines. These are described in the following sections.

Mode 0 (Basic Input/Output)

Mode 0 characteristics are as follows:

- Two 8-bit ports (A and B) and two 4-bit ports (upper and lower halves of port C).
- Any port can operate either as an input or an output.

Bit definitions for mode 0 are shown in Figure 4 - 5.

This mode of operation provides simple I/O operations. Ports defined as inputs when read reflect the digital inputs on the port. Ports defined as outputs are set to the value most recently written to the port. Any port can be used either for input or output, but not for both.

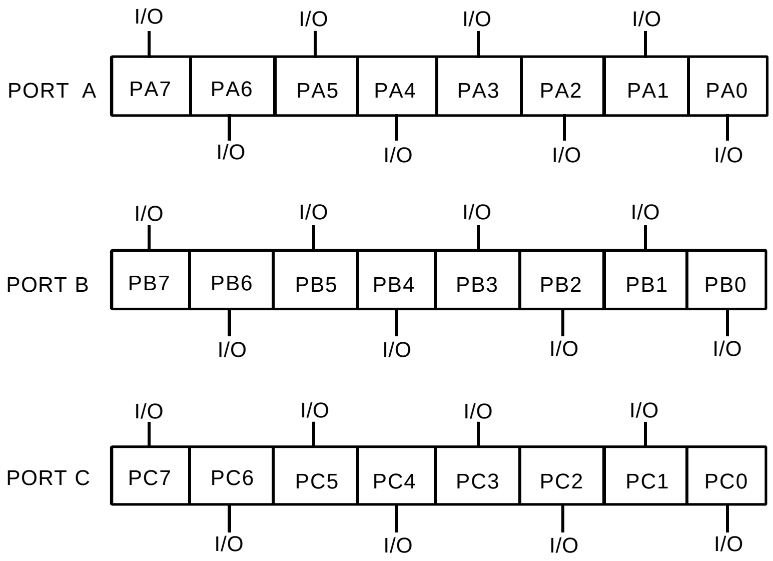


Figure 4 - 5. Mode 0 Digital I/0

Mode 1 (Strobed Input/Output)

Mode 1 characteristics are as follows:

- Two groups, each consisting of one 8-bit and one 4-bit port.
- The 4-bit port is used for control and status of the 8-bit port.
- The 8-bit port may be used for either input or output.
- Input and output operations are latched.

This mode of operation provides I/O operations with a simple handshake protocol. The assignment of handshake signals to port C is shown in Figure 4 - 6 and Figure 4 - 7. The handshake signals are as follows:

Handshake Signals

a. **Input Operations**

- **STB (Strobe input)**. A low on this input loads data into the input buffer. The data can then be read by the program.
- **IBF (Input Buffer Full)**. A high on this output indicates that there is data in the input buffer. This can be used as either an acknowledgement or buffer full signal. The output is reset when the CPU reads the data.
- **INTR (Interrupt)**. This output is set high if the STB is low, the IBF is high, and the INTE bit of the PC104-30 internal register is set.

b. **Output Operations**

- **OBF (Output Buffer Full)**. A low on this output indicates that the CPU has written data to the port. It is set high by the ACK input going low.
- **ACK (Acknowledge input)**. A low on this input indicates to the PC104-30 that the data has been accepted by the external circuitry.
- **INTR (Interrupt)**. This output is set high if the ACK is high, the OBF is high, and the INTE bit of the PC104-30 internal register is set.
- Bits of port C not used for handshake lines can be used for simple (mode 0) operations.

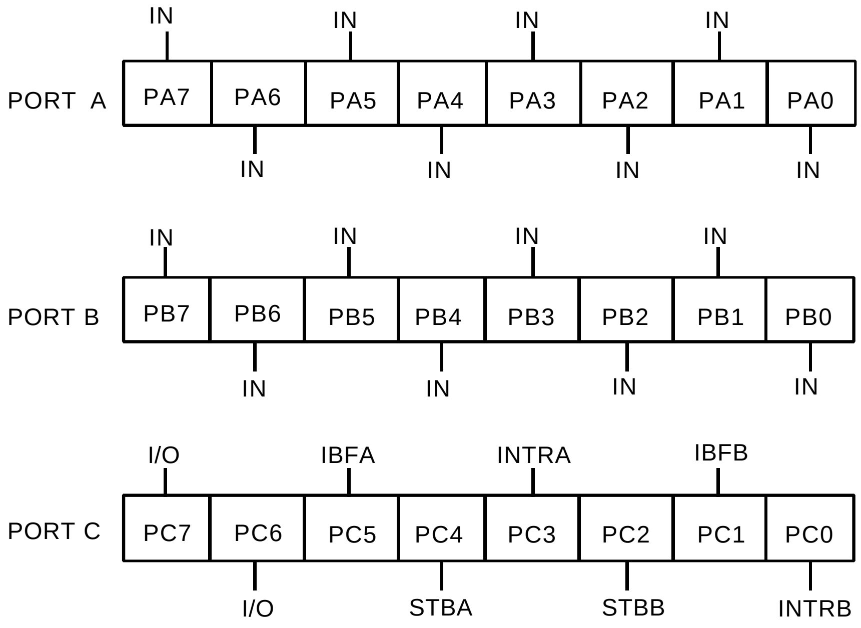


Figure 4 - 6. Mode 1 Digital Input

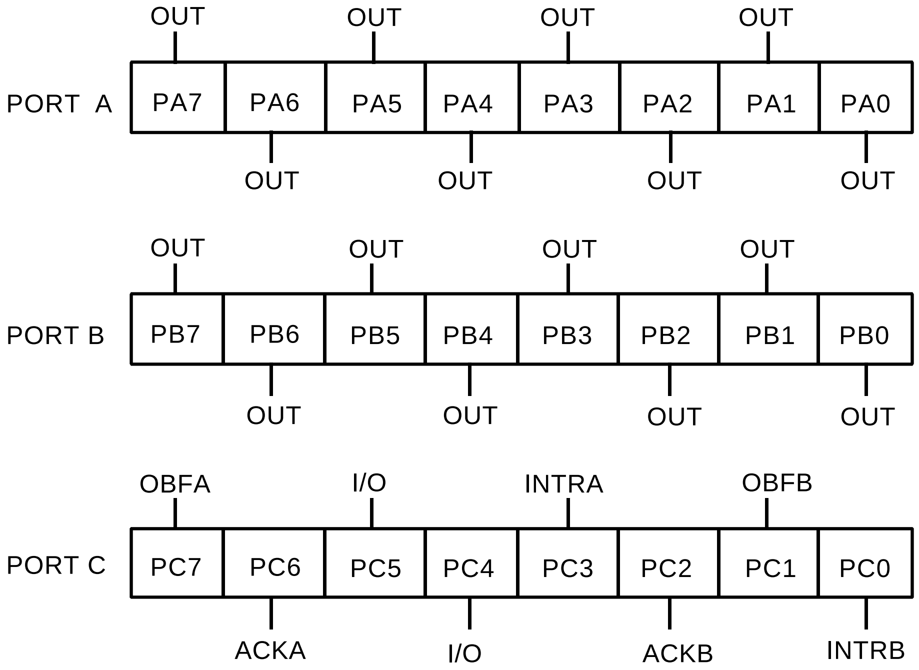


Figure 4 - 7. Mode 1 Digital Output

Mode 2 (Strobed Bi-directional Input/Output)

Mode 2 characteristics are as follows:

- One 8-bit bi-directional port, and one 5 bit control port.
- Can be used in group A only.
- The 5-bit port is used for control and status of the 8-bit port.
- Input and output operations are latched.
- Port B can still be used in mode 0 or 1.

This mode of operation provides a means for bi-directional I/O operations on port A. The assignment of handshake signals to port C is shown in Figure 4 - 8. The handshake signals are as follows:

Handshake Signals

a. **OBF (Output Buffer Full)**. A low on this output indicates that the CPU has written data to the port. It is set high by the ACK input going low.

b. **ACK (Acknowledge input)**. A low on this input enables the port A outputs, allowing external circuitry to read the value written to the port. If this output is high, port A is in the input mode.

c. **STB (Strobe input)**. A low on this input loads data into the input buffer. It can then be read by the program.

d. **IBF (Input Buffer Full)**. A high on this output indicates that there is data in the input buffer. This can be used as either an acknowledgement or buffer full signal. The output is reset when the CPU reads the data.

e. **INTR (Interrupt)**. This output is set high under either of two conditions:

- If ACK is high, OBF is high, and the INTE1 bit of the PC104-30 internal register is set.
- If STB is low, IBF is high, and the INTE2 bit of the PC104-30 internal register is set.

Bits of port C not used for handshake lines can be used for simple (mode 0) I/O operations.

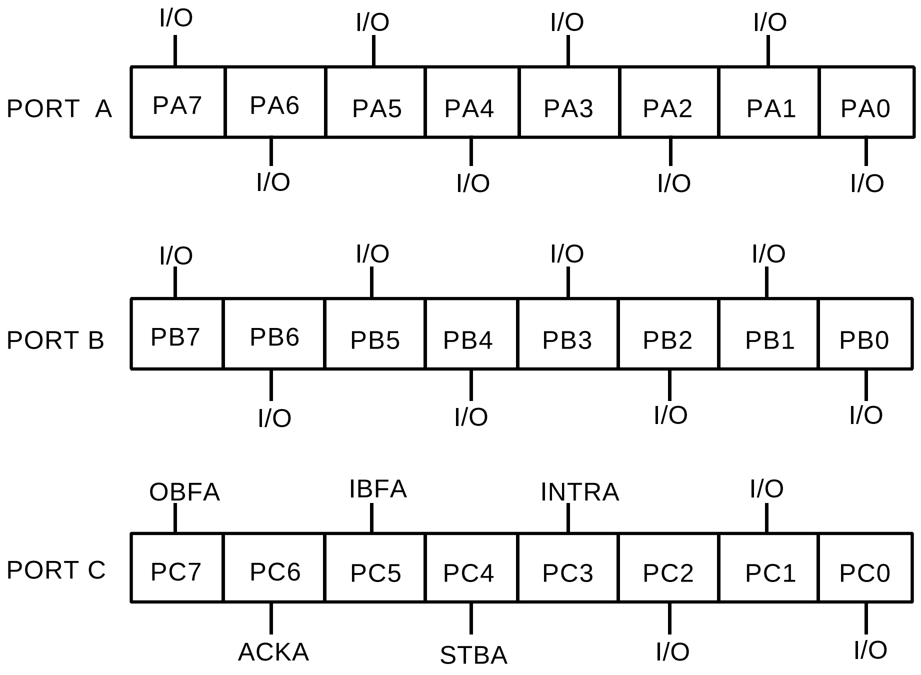


Figure 4 - 8. Mode 2 Digital I/0

Mode Combination Considerations

It is possible to configure the parallel interface of the PC104-30 in several different modes which leave some bits of port C unused for control or status. These bits can be used as follows:

- If programmed as inputs, these bits can be accessed as usual by port read commands.
- If programmed as outputs, the bits can be written by the bit set reset functions described in Chapter 5.

Connecting Normally Open devices to the PPI

When connecting switches, etc, to the Port lines (classed as inputs) of the PPI, it is important to ensure that the inputs are set to a defined state at all times. Figure below gives an example.

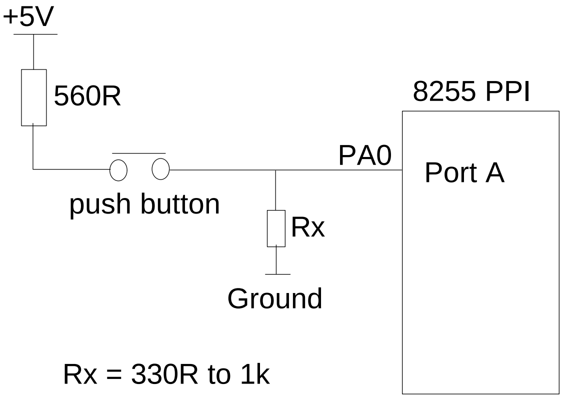


Figure 4 - 9. Connecting Normally Open Devices to the PPI

Figure above gives an example of how to connect a push button to one of the lines of the 8255 PPI. We assume that when the Push button is closed, the Port line will go high (logic 1 = +5V). When the push button is not closed, the port line should be low. However, many Users does not connect the ‘pull-down’ resistor to the port line. If this is not done, the port line will be in a floating state and can hence yield either a high or a low.

Thus you must ensure that the port line is always connected which is done via a resistor network.

Chapter 5: Register Structure

Introduction

At the lowest level, the PC104-30 can be programmed using I/O input and output instructions. This chapter contains the information required to do this. This is difficult and time consuming, and requires detailed knowledge of the PC104-30, as well as the operation of the host PC and its operating system. To simplify this process, all PC104-30 boards are supplied with our EDR Software Developers Kit. EDR provides easy access to all board functions and supports advanced functions such as streaming to disk. Read about EDR in the supplied manual.

See Chapter 6 for various programming techniques and tips.

Register Structure

The PC104-30 uses 32 consecutive address locations in I/O space. The layout of these registers is shown in Table 5-1. Note that certain addresses have different read and write register functions. This register map is compatible with older boards in the series, such as the PC104-30GA, PC30FA, etc.

Note also that the addresses are given as offsets from the base address of the board. This base address is DIP switch selected as described in Chapter 3.

Each register will now be described in detail.

CAUTION !

DO NOT WRITE TO, OR READ FROM, UNUSED REGISTERS. ALL UNUSED REGISTERS ARE RESERVED FOR MANUFACTURING TEST, OR FOR FUTURE DEVELOPMENTS.

Table 5 - 1: Register Layout

Offset From Base	Register Name	
	Read	**Write**
0	A/D Low Byte (ADDATL)	Block Count (BLKCNT)
1	A/D Data/Status (ADDSR)	Reserved
2	Control/Channel (ADCCR)	
3	A/D Mode Register (ADMDE)	
4	A/D Clock Prescaler (PRESCALER)	
5	A/D Clock Divider (DIVIDER)	
6	Uncommitted Counter/Timer (USR_CNT)	
7	-	Counter Control (TMRCTR)
8	Digital I/O Port A (DIOP0)	
9	Digital I/O Port B (DIOP1)	
10	Digital I/O Port C (DIOP2)	
11	Reserved	Dig. Control (DIOCNTRL)
12	Reserved	DAC0 Low Byte (DADATL0)
13	Reserved	DAC0 High Byte (DADATH0)
16	Reserved	DAC1 Low Byte (DADATL1)
17	Reserved	DAC1 High Byte (DADATH1)
18	DMA RQ / IRQ Mux Control (IMUX)	
19	DACK RQ / GATE Control (IGATE)	
20	Reserved	DAC2 Low Byte (DADATL2)
21	Reserved	DAC2 High Byte (DADATH2)
22	Reserved	DAC3 Low Byte (DADATL3)
23	Reserved	DAC3 High Byte (DADATH3)
24	Gain Read (GAINREG)	Gain Memory 0 (GMEM0)
25	Reserved	Gain Memory 1 (GMEM1)
26	Reserved	Gain Memory 2 (GMEM2)
27	Reserved	Gain Memory 3 (GMEM3)
28	A/D Configuration (ADCCFG)	
29	D/A Configuration (DACCFG)	
30	Clock Source Configuration (CLKSRC)	

Table 5 - 1: Register Layout

Offset From Base	Register Name	
	Read	Write
31	Reserved	Reserved

ADDATL - A/D Data Low Byte (Offset 0) (Read Only)

On completion of an A/D conversion, the A/D converter loads this register with digital data. To retrieve converted data, the user must perform a read operation to ADDATL. Bit 0 is the LSB. The layout of this register is shown in
Figure 5 - 1.

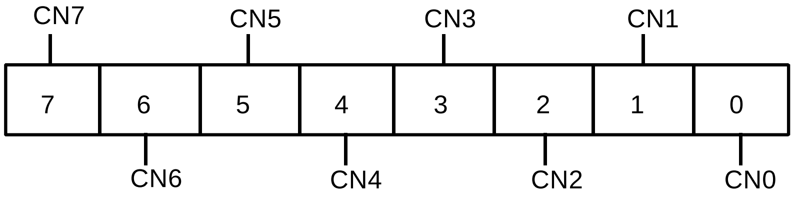


Figure 5 - 1. A/D Data Register (Low Byte)

a. Bits 7 to 0 - A/D Data (AD)

These bits are the low byte of the 12-bit code which is returned from an A/D conversion.

BLKCNT - Block Counter (Offset 0)

The block counter indicates the number of A/D conversions to perform on each A/D strobe. Note that the A/D mode must be 1. The A/D mode is set in the ADMDE register, discussed later.

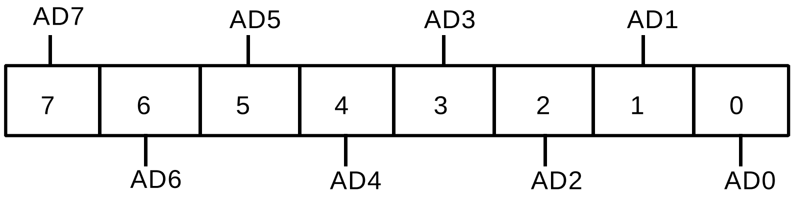


Figure 5 - 2. Block Count Register

The value written to the BLKCNT register is calculated from:

257 - (Number of conversions per block)

For example, to perform three conversions per block, the value 254 must be written to the BLKCNT register. The value written to the BLKCNT register can range from 255 to 1 (block sizes of 2 to 256). Figure 5 - 2 shows the layout of this register.

> You must write the block count register prior to writing the first value to the channel list. Failure to do this may cause unpredictable operation.

a. Bits 7 to 0 - Block Count Value (CN)

These bits represent 257 - the number of conversions to perform per clock pulse.

ADDSR - A/D Data/Status Register (Offset 1)

The ADDSR contains the high nibble of the A/D result, and contains the current A/D status information. The bit functions of the ADDSR are shown in Figure 5 - 3.

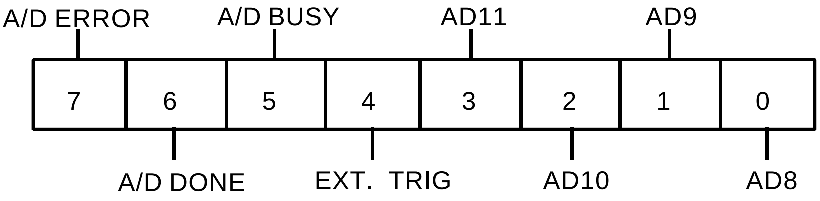


Figure 5 - 3. A/D Data/Status Register

a. Bit 7 - A/D Error

Set when an error occurs during an A/D conversion. Such conditions can result from two causes:

- The current conversion has been completed while data from the previous conversion has not been completely read (ADDATL low byte not read), which is a data overflow error,
- or, an attempt has been made to initiate a new conversion while the current conversion is still in progress, which is a trigger error.

Note that on DMA operations it is normal to have a data overflow error on completion of the operation. There should not however be any error while the transfer is in progress.

Bit 7 is cleared on power up, and by writing a 1 to bit 2 of the ADMDE register.

b. Bit 6 - A/D Done

Set by the A/D converter to indicate that A/D data is available. If bit 3 of the ADCCR is set (interrupts are enabled), then an interrupt will be generated when bit 6 is set. If bit 2 of the ADCCR is set (DMA is enabled), then a DMA cycle will be generated when bit 6 is set.

The A/D Done bit remains set as long as there is data in the FIFO buffer.

c. Bit 5 - A/D Busy

Set by an A/D strobe, and indicates that a conversion is in progress. The bit is cleared at end of conversion, and any trigger while the bit is set will cause an error condition.

d. Bit 4 - Ext. Trig

This bit reflects the status of the external trigger pin (pin 25 of the user connector).

e. Bits 3 to 0 - A/D Data (AD)

The four higher bits of 12-bit data from an A/D conversion.

ADCCR - A/D Control/Channel Register (Offset 2)

The ADCCR contains channel address bits, A/D clock control bits as well as DMA and interrupt enable bits. The bit functions of the ADCCR are shown in Figure 5 - 4.

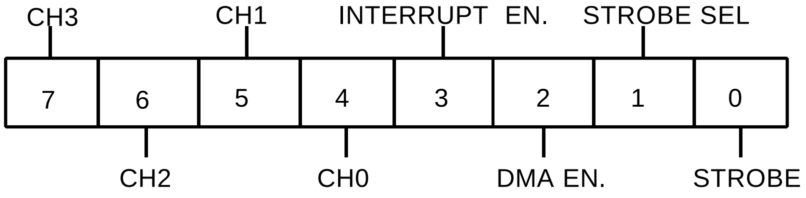


Figure 5 - 4. A/D Control/Channel Register

a. Bits 7 to 4 - Channel (CH)

These bits specify a four bit channel address. Depending on the A/D mode, the channel specified by these four bits either replaces the current channel list, is added to the current channel list, or is ignored. The A/D mode is set in the ADMDE register, discussed in the next section. The channel list is also discussed in detail in that section.

On read, these bits represent the channel at the head of the channel list, not the most recently written channel value. This channel will be converted on the next A/D strobe.

b. Bit 3 - Interrupt Enable

If the interrupt enable bit or IGATE0 (Bit 0 in the IGATE Register [Offset 19]) is set, then interrupts will be generated under one of three conditions:

(i) The end of each A/D conversion.

(ii) On each pulse from the uncommitted counter/timer.

(iii) The end of a DMA block.

(iv) External Trigger (This option will only be available on the REV1C Board.

(v) External Clock (This option will only be available on the REV 1C Board.

The selection between these conditions is software selected, as described in Chapter 3.

This bit is controlled by the program in use and is cleared on power up. All interrupts are disabled when this bit is cleared.

c. Bit 2 - DMA Enable

DMA operations are controlled by the combination of bit 2, and the DMA mode bit in the ADMDE register. This interaction of the DMA enable bit and the DMA mode bit is explained in the description of the ADMDE register.

d. Bit 1- Strobe Select (STBC)

This bit controls the source of A/D strobes to the A/D converter. If it is set, then software strobes are used. Software strobes are generated by toggling the SSTB bit (bit 0). If the STBC bit is cleared, then A/D clock pulses are used to start A/D conversion cycles. A/D clock pulses may be generated from the A/D clock prescaler/divider combination, or from an external source.

e. Bit 0 - Strobe (SSTB)

Bit 0 of the ADCCR is the software strobe bit. If the STBC bit is set, then a software strobe is generated by taking the strobe bit high, then low. If the STBC bit is low, then the SSTB bit is ignored.

CAUTION

WHENEVER YOU CHANGE THE STATE OF THE STBC BIT, THE SSTB BIT MUST BE 0 (CLEARED). IF THE SSTB BIT IS NOT CLEAR, SPURIOUS A/D CONVERSION CYCLES MAY BE GENERATED.

ADMDE - A/D Mode Register (Offset 3)

The ADMDE register contains A/D mode selection bits, the error reset control bit and the DMA mode control bit. The bit functions of the ADMDE register are shown in Figure 5 - 5.

If your application requires 100% compatibility with older PC26, PC104-30 or PC39 boards, you should write the value 92 (hex) to this register.

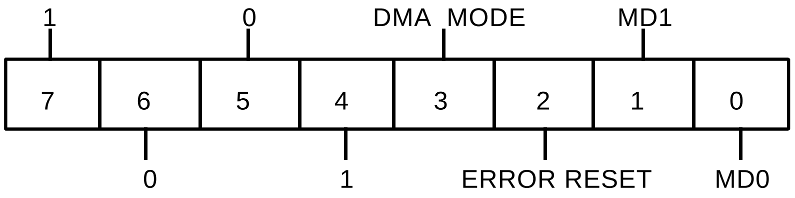


Figure 5 - 5. A/D Mode Register

a. **Bit 7 - Reserved (1)**

   For compatibility with future products, you must write a 1 to this bit. The results of reading this bit are undefined.

b. **Bit 6 - Reserved (0)**

   For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

c. **Bit 5 - Reserved (0)**

   For compatibility with future products, you must write a 0 to this bit. The results of reading this bit are undefined.

d. **Bit 4 - Reserved (1)**

   For compatibility with future products, you must write a logical 1 to this bit. The results of reading this bit are undefined.

e. **Bit 3 - DMA Mode**

   The table below describes the effect of the various combinations of the DMA enable bit in the ADCCR. There are four possible states:

Enable Bit	Mode Bit	Description
0	0	**Disable DMA**. The PC104-30 DMA request lines are held tri-state.
0	1	**Swap on TC**. This mode is used during dual channel DMA. On each TC pulse, the DMA enable bit is set, and the PC104-30's internal block flip-flop is toggled. The PC104-30 hence begins DMA on the next DMA channel.
1	0	**Normal DMA**. This mode can be used for normal single channel DMA. TC pulses are ignored. This mode can be used for "circular buffer" applications where DMA runs continuously.
1	1	**Terminate on TC**. The PC104-30 switches itself to this state from the 0-1 state. On the next TC pulse, all DMA activities are halted. This is the recommended mode for single channel DMA transfers.

End of DMA block interrupts are produced by the DMA block toggle signal described above. Hence, to generate DMA interrupts, the PC104-30 must be programmed for "Swop on TC" mode operation.

f. Bit 2 - Error Reset

Writing a 1 to this bit clears the error detection bit in the ADDSR.

g. Bits 1 and 0 - Mode (MD)

The two mode bits set the A/D conversion mode. The A/D conversion mode sets the channel list mode, selects the trigger mode and enables or disables the FIFO buffer. The following table details the effect of the various modes.

Mode Bits		**Channel List Mode**	**Trigger Mode**	**FIFO Mode**
MD1	**MD0**			
0	0	Ignored	Normal	Enabled
0	1	Ignored	Burst	Enabled
1	0	Replace	Normal	Disabled
1	1	Add	-	Disabled

i. **Channel List Mode**

There are three possible channel list modes: replace, ignore and add.

- **Replace**. In replace mode, the current channel list is cleared, and the channel value written to the ADCCR register is inserted into the head of the list. This entry is then the only value in the channel list.
- **Ignore**. In ignore mode, the channel bits in the ADCCR register have no effect. This is used to change control bits (for example the DMA and interrupt enable bits) without modifying the current channel list.
- **Add**. In add mode, the channel value written is added to the current channel list.

ii. **Trigger Mode**. Trigger mode was discussed extensively in Chapter 2. Either normal or burst (block) mode can be selected. Note that in block mode, simultaneous sample and hold operation is active.

iii. **FIFO Mode**. The FIFO buffer can be either enabled or disabled. Note that disabling the FIFO buffer resets it, destroying all data currently stored in it.

PRESCALER - A/D Clock Prescaler Register (Offset 4)

This register is used to program the A/D clock prescaler. Before it can be used, the counter must be configured by writing the appropriate mode setting byte to the counter control register, TMRCTR, described later. The counter can be configured either as a binary or BCD counter, although binary mode is normally used.

The counter can be configured into several operating modes, as discussed in the description of the TMCTR register. Default mode setting is mode 2.

The input to the prescaler is set to either the 2 MHz (default) or 8 MHz master clock of the PC104-30.

This register is register 0 (counter 0) of the 8254 on the PC104-30 board. The bit layout of the prescaler is shown in Figure 5 - 6.

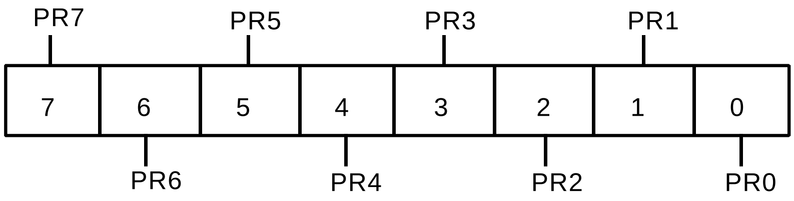


Figure 5 - 6. Prescaler Register

a. Bits 7 to 0 - Prescaler Data (PR)

The prescaler register can be configured to read/write either the high byte of the counter, the low byte of the counter, or both bytes in sequence. The normal configuration is to write both bytes. In this case the LSB is written first, then the MSB.

It is important to always read/write two bytes from the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

DIVIDER - A/D Clock Divider Register (Offset 5)

The divider register is used to program the A/D clock divider. Before it can be used, the counter must be configured by writing the appropriate mode setting byte to the counter control register, TMRCTR, described later. The counter can be configured either as a binary or BCD counter, although binary mode is normally used.

The counter can be configured into several operating modes, as discussed in the description of the TMCTR register. Default mode setting is mode 2.

The input to the clock divider is always the output from the A/D clock prescaler.

This register is register 1 (counter 1) of the 8254 on the PC104-30 board. The bit layout of the divider shown in Figure 5 - 7.

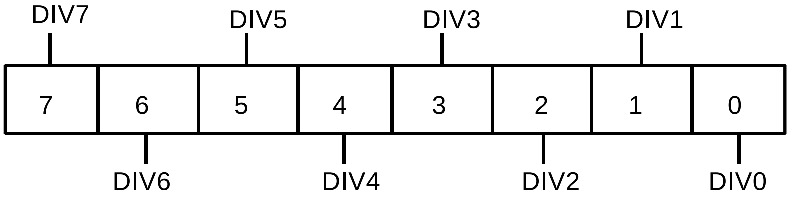


Figure 5 - 7. A/D Clock Divider Register

a. Bits 7 to 0 - A/D Clock Divider Data (DIV)

The clock divider register can be configured to read/write either the high byte of the counter, the low byte of the counter, or both bytes in sequence. The normal configuration is to write both bytes. In this case the LSB is written first, then the MSB.

It is important to always read/write two bytes from the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

USR_CNT - User Counter Register (Offset 6)

This register is used to program and to read the user configurable counter/timer. Before it can be used, the counter must be configured by writing the appropriate mode setting byte to the counter control register, TMRCTR, described later. The counter can be configured either as a binary or BCD counter, although binary mode is normally used. The clock and gate source for the counter should also be configured using the CLKSRC register (Offset 30).

The counter can be configured into several operating modes, as discussed in the description of the TMCTR register. Default mode setting is mode 2.

This register is register 2 (counter 2) of the 8254 on the PC104-30 board. The bit layout of the USR_CNT register is shown in Figure 5 - 8.

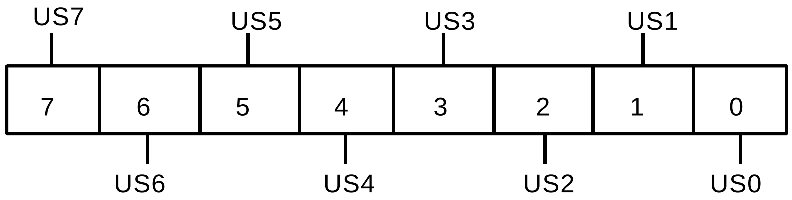


Figure 5 - 8. User Counter Register

a. Bits 7 to 0 - User Counter Data (US)

The user counter register can be configured to read/write either the high byte of the counter, the low byte of the counter, or both bytes in sequence. The normal configuration is to read both bytes. In this case the LSB is read first, then the MSB.

It is important to always read/write two bytes from the counter if it is configured for 16-bit operation. Failing to do this can result in invalid data.

TMRCTR - Timer Control Register (Offset 7)

The timer control register is used to configure the three 16-bit counters on the PC104-30 board. It is register 3 (Mode word) of the 8254 on the board. If you intend to program the 8254 extensively, you should obtain a copy of the data sheet for an 8254 type counter/timer.

The register layout is shown in Figure 5 - 9.

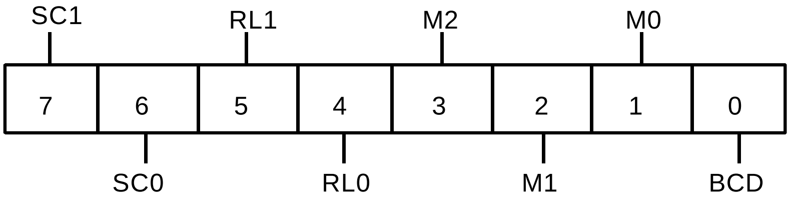


Figure 5 - 9. Timer Controller Register

a. Bits 7 and 6 - Select Counter (SC)

These two bits are used to select the counter to be configured and function as follows:

SC1	SC0	Function
0	0	Select counter 0 (A/D clock prescaler). The rest of the information in the byte written is used to configure the prescaler.
0	1	Select counter 1 (A/D clock divider). The rest of the information in the byte written is used to configure the divider.
1	0	Select counter 2 (User configured counter/timer). The rest of the information in the byte written is used to configure the user counter.
1	1	Reserved; Do not use.

b. Bits 5 and 4 - Read/Load (RL)

Bits 5 and 4 are used to configure how the counter is read and written. The meaning of the various bit combinations is as follows:

RL1	RL0	Function
0	0	**Counter Latch**. The selected counter is latched. This is used to read the contents of a counter while the clock of the counter in question is active. If you intend to use this function, consult the 8254 data sheet for full details.
0	1	**Read/write LSB only**. All read/write operations will read/ write only the LSB (least significant byte) of the selected counter.
1	0	**Read/write MSB only**. All read/write operations will read/write only the MSB (most significant byte) of the selected counter.
1	1	**Read LSB/MSB**. Both the LSB and MSB of the counter are read/written. The LSB is read/written first, then the MSB. Note that both bytes must always be read/written. Reading/writing only one byte will cause unpredictable results.

c. Bits 3 to 1 - Mode (M)

The counters can be programmed into various modes, as described below. The normal mode of operation for counters 0 and 1 (the A/D prescaler and divider) is mode 2. The mode for counter 2 (the user configurable counter) depends on its intended application.

Mode Bits (M2 M1 M0)	**Mode Description**
0 0 0	Mode 0, interrupt on terminal count. After the mode byte is written, the output is low. Once a count value is written, the output remains low until the counter counts down to 0. The output then goes high, and remains high until a new count or mode is written to the counter. The gate input of the counter disables counting when low. This mode can be used to generate a positive edge on the external output after a programmable time or, if the board is jumpered for timer interrupts, to generate an interrupt after a programmable time. Mode 0 is also used to count events or frequency.
0 0 1	Mode 1, programmable one shot. Any rising edge on the clock input to the counter causes the output of the counter to go low for the number of clock cycles programmed into the counter. This mode can be used to generate a pulse of programmable length to external circuitry on A/D conversion.
0 1 0	Mode 2, rate generator. The output of the counter goes low for one clock period in every N clocks, where N is the number programmed into the counter. This is the normal mode for the A/D prescaler and clock divider. The gate input of the counter disables counting when low. This mode can also be used in the configurable counter to generate an output frequency, or to generate a periodic interrupt.
0 1 1	Mode 3, square wave generator. The output of the counter goes low for N/2 clocks in every N, where N is the number programmed into the counter. The counter hence generates square waves. The gate input of the counter disables counting when low. This mode is normally used in the uncommitted timer to generate an output frequency. Note that the minimum value of N is four.

Mode Bits (M2 M1 M0)	**Mode Description**
1 0 0	Mode 4, software triggered strobe. After the mode byte or a count value is written, the output is high. The output remains high until the counter counts down to 0. The output then goes low for one clock period. The gate input of the counter disables counting when low. This mode can be used to generate a pulse on the external output after a programmable time.
1 0 1	Mode 5, hardware triggered strobe. After the mode byte or a count value is written, the output is high. The counter begins counting down after a rising edge on the gate input. When the count reaches 0, the output goes low for one clock period. This mode can be used to generate a pulse on the external output after a programmable time.

d. Bit O - BCD

If this bit is 0, the counter is configured as a binary counter. If it is 1, the counter is configured as a BCD counter. Note that in either case the counter is a down counter.

DIOP0 - Digital I/O Port 0 (Offset 8)

This register is digital I/O port 0 (port A). It can be operated in one of several modes, as discussed in Chapter 4. The mode is set in the DIOCNTRL register, described below. The layout of the DIOP0 register is shown in Figure 5 - 10.

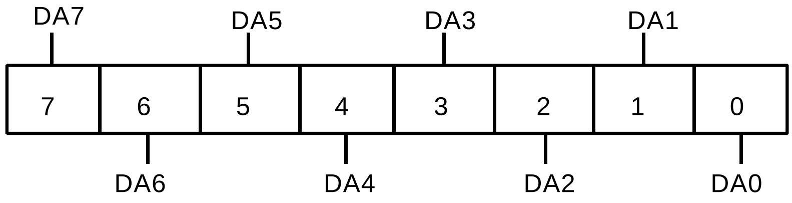


Figure 5 - 10. Digital I/O Port 0 Register

a. Bits 7 to 0 - Digital I/O Port 0 (DA)

These bits are port 0 of the 8255 on the PC104-30 board. Depending on the port mode, the bits reflect the status of the I/O lines of the port in question. These lines may be inputs, outputs, bi-directional or handshake lines, depending on mode. The various modes are discussed in detail in Chapter 4, and in the description of the DIOCNTRL register.

DIOP1 - Digital I/O Port 1 (Offset 9)

This register is digital I/O port 1 (port B). It can be operated in one of several modes, as discussed in Chapter 4. The mode is set in the DIOCNTRL register, described below. The layout of the DIOP1 register is shown in Figure 5 - 11.

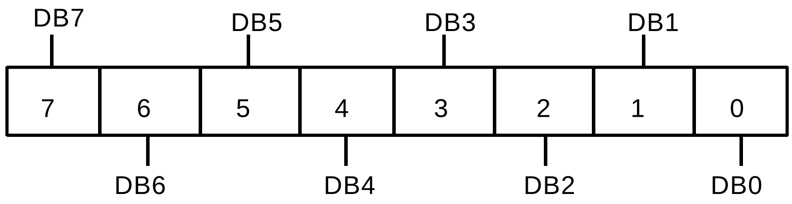


Figure 5 - 11. Digital I/O Port 1 Register

a. **Bits 7 to 0 - Digital I/O Port 1 (DB)**

These bits are port 1 of the 8255 on the PC104-30 board. Depending on the port mode, the bits reflect the status of the I/O lines of the port in question. These lines may be inputs, outputs, bidirectional or handshake lines, depending on mode. The various modes are discussed in detail in Chapter 4, and in the description of the DIOCNTRL register.

DIOP2 - Digital I/O Port 2 (Offset 10)

This register is digital I/O port 2 (port C). It can be operated in one of several modes, as discussed in Chapter 4. The mode is set in the DIOCNTRL register, described below. The layout of the DIOP2 register is shown in Figure 5 - 12.

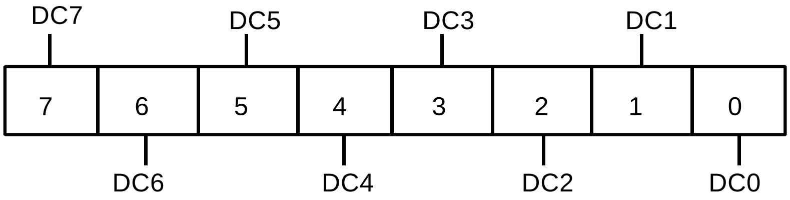


Figure 5 - 12. Digital I/O Port 2 Register

a. **Bits 7 to 0 - Digital I/O Port 2 (DC)**

These bits are port 2 of the 8255 on the PC104-30 board. Depending on the port mode, the bits reflect the status of the I/O lines of the port in question. These lines may be inputs, outputs, bi-directional or handshake lines, depending on mode. The various modes are discussed in detail in Chapter 4, and in the description of the DIOCNTRL register.

DIOCNTRL - Digital I/O Control (Offset 11)

This register can either be used to control the mode of the three digital I/O ports, or to set and reset individual bits in port C. This is the control register of the 8255 on the PC104-30 board. The various possible modes were described in Chapter 4. If you intend to program the 8255 extensively, you should also obtain a data sheet for an 8255 type device for further information.

The layout of this register is shown in Figure 5 - 13. Note that the register function and layout depends on the setting of bit 7.

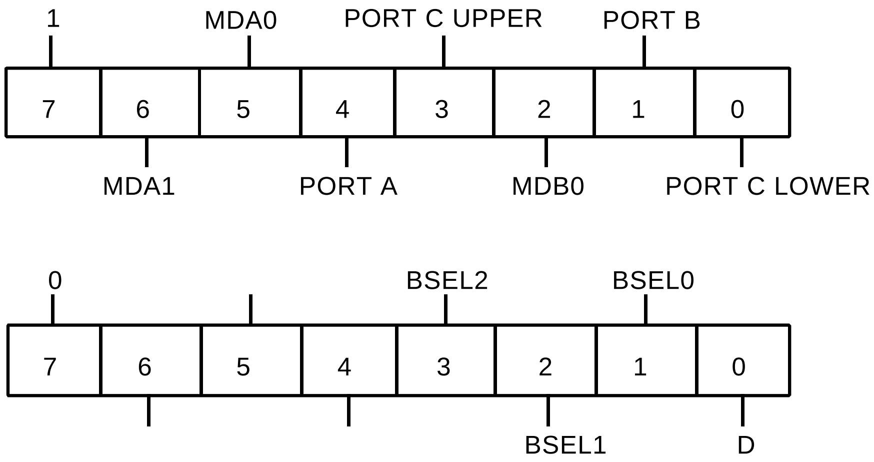


Figure 5 - 13. Digital I/0 Control Register

a. Bit 7 - Function Select

If this bit is 1, the register is in configuration mode. If the bit is 0, then it is in bit set/reset mode.

Configuration Mode

a. Bits 6 and 5 - Group A Mode Set (MDA)

These two bits set the mode of the group A ports. These are port A, and the upper nibble of port C. The bit combinations are as follows:

MDA1	MDA0	Mode
0	0	Mode 0, simple I/O.
0	1	Mode 1, strobed I/O.
1	0	Mode 2, strobed bi-directional I/O.
1	1	Reserved; do not use.

b. Bit 4 - Port A Input

If this bit is set, then port A functions as an input. If it is zero, then port A is an output.

c. Bit 3 - Port C Upper Input

If this bit is set, then the group A portion of port C (the upper nibble) functions as an input. If it is zero, then the upper nibble of port C is an output.

d. Bit 2 - Group B Mode Set (MDB)

This bit sets the mode of the group B ports. These are port B, and the lower nibble of port C. The bit settings are as follows:

MDB0	Mode
0	Mode 0, simple I/O.
1	Mode 1, strobed I/O.

e. Bit 1- Port B Input

If this bit is set, then port B functions as an input. If it is zero, then port B is an output.

f. Bit 0 - Port C Lower Input

If this bit is set, then the group B portion of port C (the lower nibble) functions as an input. If it is zero, then the lower nibble of port C is an output.

Bit Set/Reset Mode

a. Bits 6 to 4 - Reserved

The content of these bits is ignored.

b. Bits 3 to 1 - Bit Select (BSEL)

The BSEL bits serve to select the bit in port C which is to be modified. A code of 000 selects bit 0, a code of 001 selects bit 1, etc.

c. Bit 0 - Data (D)

The data bit represents the value to which the selected bit will be set.

DADATL0 - DAC0 Register (Low Byte) (Offset 12)

This register is used to hold the four lower bits of the 12-bit code loaded into DAC0 by software for D/A conversions. Data is left justified and is transferred to the output when this register is written. Figure 5 - 14 shows the register layout.

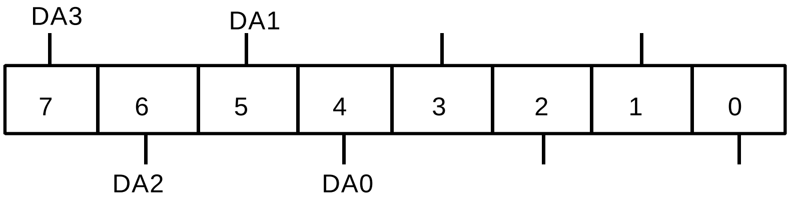


Figure 5 - 14. DAC0 Low Byte Data Register

a. Bits 7 to 4 - DAC0 Data (DA)

These bits are the LSB of DAC0 data.

b. Bits 3 to 0 - Not Used

DADATH0 - DAC0 Register High Byte (Offset 13)

DADATH0 high byte holds the eight higher bits of the software-loaded 12-bit code for D/A conversion. Bit 7 is the MSB. Data is left justified. Note that changes to this register are not reflected in the output until the low byte register is written. The layout of this register is shown in Figure 5 - 15.

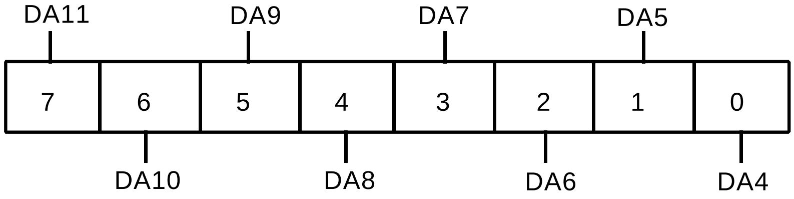


Figure 5 - 15. DAC0 High Byte Data Register

a. Bits 7 to 0 - DAC0 Data (DA)

These eight bits are the MSB of the DAC0 data.

DADATL1 -DAC1 Register (Low Byte) (Offset 16)

This register is used to hold the four lower bits of the 12-bit code loaded into DAC1 by software for D/A conversions. Data is left justified and is transferred to the output when this register is written. Figure 5 - 16 shows the register layout.

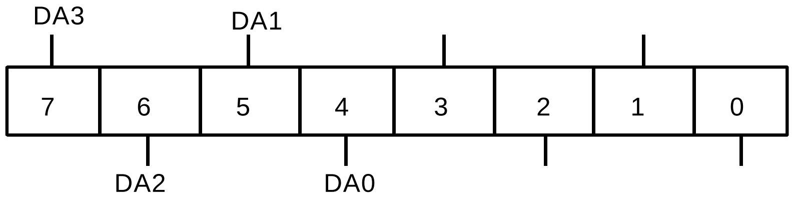


Figure 5 - 16. DACl Low Byte Data Register

a. Bits 7 to 4 - DAC1 Data (DA)

These bits are the LSB of DAC1 data.

b. Bits 3 to 0 - Not Used

DADATH1 - DAC1 Register High Byte (Offset 17)

DADATH1 high byte holds the eight higher bits of the software-loaded 12-bit code for D/A conversion. Bit 7 is the MSB. Data is left justified. Note that changes to this register are not reflected in the output until the low byte register is written. The layout of this register is shown in Figure 5 - 17.

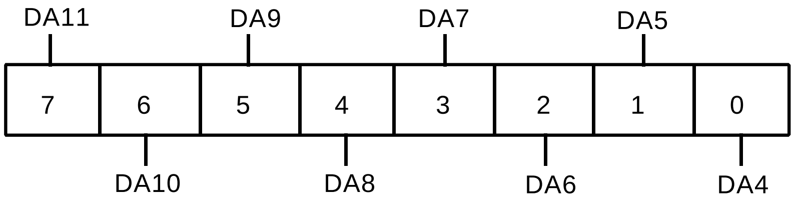


Figure 5 - 17. DACl High Byte Data Register

a. Bits 7 to 0 - DAC1 Data (DA)

These eight bits are the MSB of the DAC1 data.

IMUX – DMA/IRQ Config Register (Offset 18)

The IMUX Register in conjunction with the IGATE Register (Offset 19) is used to select the DMA request Channel and the IRQ Channel. The bit functions of the IMUX are shown in Figure 5 - 3.

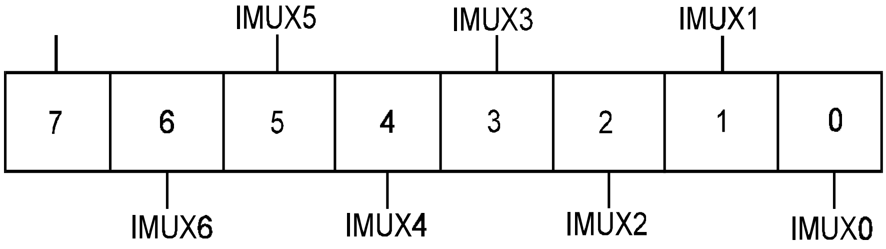


Figure 5 - 18. A/D Data/Status Register

a. **Bit 7 - Reserved**

Reserved. In order to maintain compatibility, write a 0 to this bit. Reading this bit is undefined.

b. **Bit 6 to 5 – IMUX6 to IMUX5**

These bits in conjunction with Bit 2 of the IGATE Register [Offset 19] is used to select the DMA request Lines of the Secondary DMA Channel. The Table below describes the bit combinations required to set the Secondary DRQ Channels:

IGATE2	**IMUX6**	**IMUX5**	**DRQ B**
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	X
0	X	x	Tri-State

Note that the Power Up default is no DMA Channels selected. In other words the DMA Channel is tri-stated (Not connect) from the BUS.

c. **Bit 4 to 3 – IMUX 4 to IMUX3**

These bits in conjunction with Bit 1 of the IGATE Register [Offset 19] is used to select the DMA request Lines of the Primary DMA Channel. The Table below describes the bit combinations required to set the Primary DRQ Channels:

IGATE1	**IMUX4**	**IMUX3**	**DRQ A**
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	X
0	X	X	Tri-State

Note that the Power Up default is no DMA Channels selected. In other words the DMA Channel is tri-stated (Not connect) from the BUS.

d. Bits 2 to 0 – IMUX2 to IMUX0

These bits in conjunction with Bit 0 of the IGATE Register [Offset 19] is used to select the Interrupt Request Lines. The Table below describes the bit combinations required to set the Interrupt Request Line:

IGATE0	IMUX2	IMUX1	IMUX0	IRQ
1	0	0	0	2
1	0	0	1	7
1	1	1	0	5
1	1	1	1	3
1	1	0	0	10
1	1	0	1	11
1	1	1	0	12
1	1	1	1	15
0	X	X	X	Tri-State

Note that the Power Up default is no IRQ Channel selected. In other words the IRQ Line is tri-stated (Not connect) from the BUS.

IGATE – DACK/Gate Control Register (Offset 19)

The DAC/GATE Register in conjunction with the IMUX Register (Offset 18) is used to select the DMA request Channel and the IRQ Channel. Furthermore, the DACK/GATE register also configures the Primary/Secondary DMA Acknowledge Lines. The bit functions of the IGATE Register are shown in Figure 5 - 3.

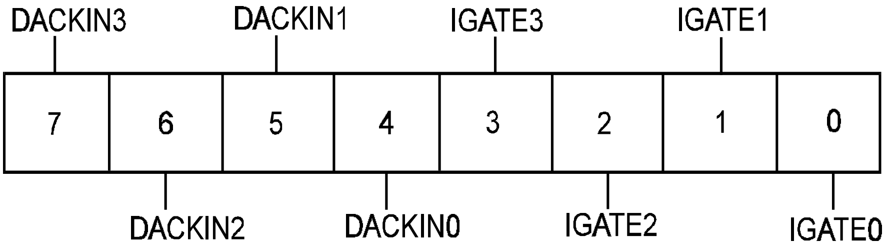


Figure 5 - 19. A/D Data/Status Register

a. Bit 7- Reserved

Reserved. In order to maintain compatibility, write a 0 to this bit. Reading this bit is undefined.

b. Bit 6 to 5- DACKIN3 to DACKIN2

These bits are used to set the DACK Input Lines of the Secondary DMA Channel. The Table below describes the bit combinations required to set the Secondary DACK Channels:

DACKIN3	**DACKIN2**	**2nd DACK CHAN**
0	0	N/C
0	1	DACK5
1	0	DACK4
1	1	DACK3

Note that the Power Up default sets DACKIN3 and DACKIN2 to 0. This implies that none of the DACK Channels are selected.

c. Bit 4 to 3- DACKIN1 to DACKIN0

These bits are used to set the DACK Input Lines of the Primary DMA Channel. The Table below describes the bit combinations required to set the Primary DACK Channels:

DACKIN1	**DACKIN0**	**1st DACK CHAN**
0	0	N/C
0	1	DACK5
1	0	DACK4
1	1	DACK3

Note that the Power Up default sets DACKIN1 and DACKIN0 to 0. This implies that none of the DACK Channels are selected.

d. Bit 2 – IGATE2

This bit in conjunction with Bits 5 and 6 of the IMUX Register [Offset 18] is used to select the DMA Request lines of the Secondary DMA Channel. The Table below describes the bit combinations required to set the Secondary DMA Channel:

IGATE2	**IMUX6**	**IMUX5**	**DRQ B**
1	0	0	5

1	0	1	6
1	1	0	7
1	1	1	X
0	X	x	Tri-State

Note that the Power Up default on IGATE2 is 0 which disables DMA Channel selection. In other words the DMA Channel is tri-stated (Not connect) from the BUS.

e. Bit 1 – IGATE1

This bit in conjunction with Bits 4 and 3 of the IMUX Register [Offset 18] is used to select the DMA Request lines of the Primary DMA Channel. The Table below describes the bit combinations required to set the Primary DMA Channel:

IGATE1	**IMUX4**	**IMUX3**	**DRQ A**
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	X
0	X	x	Tri-State

Note that the Power Up default sets IGATE1 to 0 which disables DMA Channel selection. In other words the DMA Channel is tri-stated (Not connect) from the BUS.

f. Bit 0 – IGATE0

This bit in conjunction with Bits 2 to 0 of the IMUX Register [Offset 18] is used to select the Interrupt Request lines. The Table below describes the bit combinations required to set the Interrupt Request Lines:

IGATE0	IMUX2	IMUX1	IMUX0	IRQ
1	0	0	0	2
1	0	0	1	7
1	1	1	0	5
1	1	1	1	3
1	1	0	0	10
1	1	0	1	11
1	1	1	0	12
1	1	1	1	15
0	X	X	X	Tri-State

Note that the Power Up default sets IGATE0 to 0. This disables the IRQ Channels from being selected. In other words the IRQ Line is tri-stated (Not connect) from the BUS.

DADATL2 –DAC2 Register (Low Byte) (Offset 20)

This register is used to hold the four lower bits of the 12-bit code loaded into DAC2 by software for D/A conversions. Data is left justified and is transferred to the output when this register is written. Figure 5 - 16 shows the register layout.

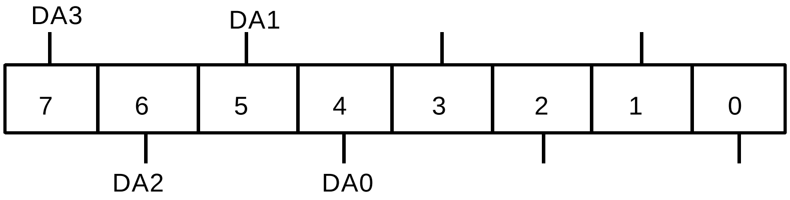


Figure 5 - 20. DAC2 Low Byte Data Register

b. Bits 7 to 4 – DAC2 Data (DA)

These bits are the LSB of DAC2 data.

c. Bits 3 to 0 - Not Used

DADATH2 – DAC2 Register High Byte (Offset 21)

DADATH2 high byte holds the eight higher bits of the software-loaded 12-bit code for D/A conversion. Bit 7 is the MSB. Data is left justified. Note that changes to this register are not reflected in the output until the low byte register is written. The layout of this register is shown in Figure 5 - 17.

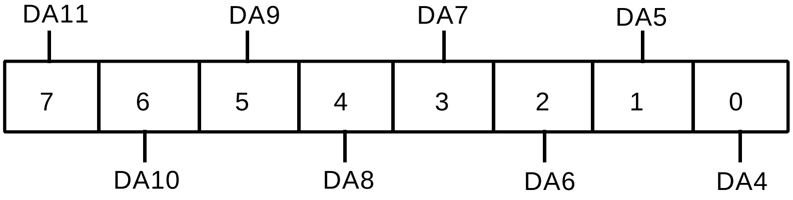


Figure 5 - 21. DAC2 High Byte Data Register

b. Bits 7 to 0 – DAC2 Data (DA)

These eight bits are the MSB of the DAC2 data.

DADATL3 -DAC1 Register (Low Byte) (Offset 22)

This register is used to hold the four lower bits of the 12-bit code loaded into DAC3 by software for D/A conversions. Data is left justified and is transferred to the output when this register is written. Figure 5 - 16 shows the register layout.

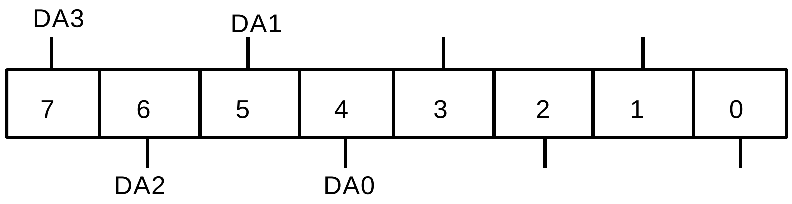


Figure 5 - 22. DAC3 Low Byte Data Register

c. Bits 7 to 4 – DAC3 Data (DA)

These bits are the LSB of DAC3 data.

d. Bits 3 to 0 - Not Used

DADATH3 – DAC3 Register High Byte (Offset 23)

DADATH3 high byte holds the eight higher bits of the software-loaded 12-bit code for D/A conversion. Bit 7 is the MSB. Data is left justified. Note that changes to this register are not reflected in the output until the low byte register is written. The layout of this register is shown in Figure 5 - 17.

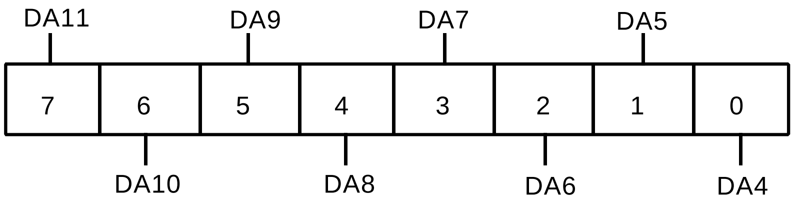


Figure 5 - 23. DAC3 High Byte Data Register

c. Bits 7 to 0 - DAC3 Data (DA)

These eight bits are the MSB of the DAC3 data.

GAINREG - Gain Read Back and Board Type Register (Offset 24)

The GAINREG register reflects the gain setting for the current input channel and contains bits that may be used to determine the model of the board. The current input channel can be found by reading the ADCCR register. The layout of this register is shown in Figure 5 - 24. Note that only the F, FA, G and GA models support programmable gain.

Channel gain is set in the GMEM0, GMEM1, GMEM2 and GMEM3 registers.

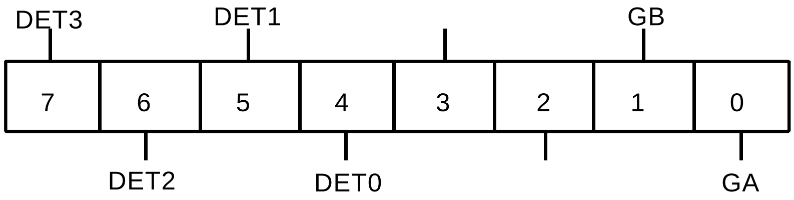


Figure 5 - 24. GAINREG Register

a. Bits 1 and 0 - Gain (GA and GB)

These bits reflect the gain setting of the current channel.

b. Bits 3 and 2 - Unused

Bits 3 and 2 are not used. The result of reading these bits is undefined.

c. Bits 7 to 4 - Model (DET3, DET2, DET1 AND DET0)

These bits can be used to determine what model of F/G is present as shown in the following table.

Board Model	**DET3-0**
Reserved	0000
PC104-30GA	0001
PC104-30G	0010
Reserved	0011
Reserved	0100
Reserved	0101
Reserved	0110
PC104-30FA	0111
PC104-30F	1000
Reserved	1001
Reserved	1010
Reserved	1011
Reserved	1100
Reserved	1101
Reserved	1110
Reserved	1111

GMEM0 - Gain Memory 0 Register (Offset 24)

The GMEM0 register holds the programmable gain settings for channels 0, 4, 8 and 12. The layout of this register is shown in Figure 5 - 25.

For each channel, the gain is set by two bits, GA and GB, as follows:

GB	GA	Gain Setting
0	0	1
0	1	10
1	0	100
1	1	1000

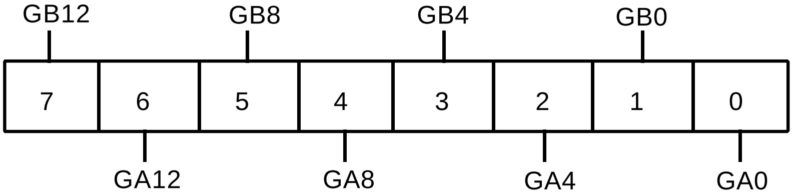


Figure 5 - 25. GMEM0 Register

The gain of the current channel can be read back from the GAINREG register.

a. Bits 1 and 0 - Channel 0 Gain (GB0 and GA0)

Channel 0 gain setting bits.

b. Bits 3 and 2 - Channel 4 Gain (GB4 and GA4)

Channel 4 gain setting bits.

c. Bits 5 and 4 - Channel 8 Gain (GB8 and GA8)

Channel 8 gain setting bits.

d. Bits 1 and 0 - Channel 12 Gain (GB12 and GA12)

Channel 12 gain setting bits.

GMEM1 - Gain Memory 1 Register (Offset 25)

The GMEM1 register holds the programmable gain settings for channels 1, 5, 9 and 13. The layout of this register is shown in Figure 5 - 26.

Gain codes are discussed in the description of the GMEM0 register.

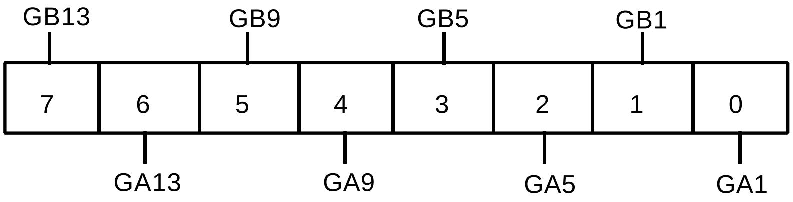


Figure 5 - 26. GMEM1 Register

a. **Bits 1 and 0 - Channel 1 Gain (GB1 and GA1)**

   Channel 1 gain setting bits.

b. **Bits 3 and 2 - Channel 5 Gain (GB5 and GA5)**

   Channel 5 gain setting bits.

c. **Bits 5 and 4 - Channel 9 Gain (GB9 and GA9)**

   Channel 9 gain setting bits.

d. **Bits 1 and 0 - Channel 13 Gain (CB13 and GA13)**

   Channel 13 gain setting bits.

GMEM2 - Gain Memory 2 Register (Offset 26)

The GMEM2 register holds the programmable gain settings for channels 2, 6, 10 and 14. The layout of this register is shown in Figure 5 - 27.

Gain codes are discussed in the description of the GMEM0 register.

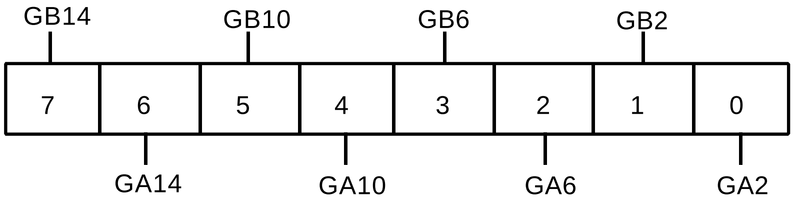


Figure 5 - 27. GMEM2 Register

a. **Bits 1 and 0 - Channel 2 Gain (GB2 and GA2)**

   Channel 2 gain setting bits.

b. **Bits 3 and 2 - Channel 6 Gain (GB6 and GA6)**

   Channel 6 gain setting bits.

c. **Bits 5 and 4 - Channel 10 Gain (GB10 and GA10)**

   Channel 10 gain setting bits.

d. **Bits 1 and 0 - Channel 14 Gain (GB14 and GA14)**

   Channel 14 gain setting bits.

GMEM3 - Gain Memory 3 Register (Offset 27)

The GMEM3 register holds the programmable gain settings for channels 3, 7, 11 and 15. The layout of this register is shown in Figure 5 - 28.

Gain codes are discussed in the description of the GMEM0 register.

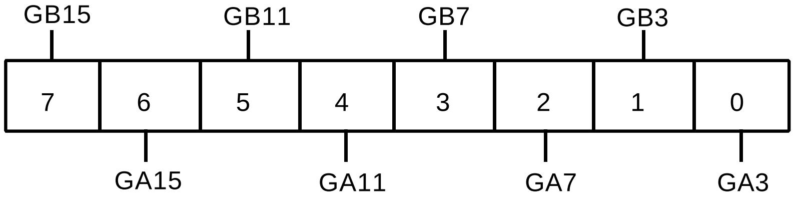


Figure 5 - 28. GMEM3 Register

a. **Bits 1 and 0 - Channel 3 Gain (GB3 and GA3)**

   Channel 3 gain setting bits.

b. **Bits 3 and 2 - Channel 7 Gain (GB7 and GA7)**

   Channel 7 gain setting bits.

c. **Bits 5 and 4 - Channel 11 Gain (GB11 and GA11)**

   Channel 11 gain setting bits.

d. **Bits 7 and 6 - Channel 15 Gain (GB15 and GA15)**

   Channel 15 gain setting bits.

Configuration Registers

These registers replace many of the jumpers found on the older series boards. All these registers default to 0 on power up or reset unless otherwise specified.

ADCCFG - A/D Configuration Register (Offset 28)

The ADCCFG register holds the interrupt source, input signal range and input signal mode. The register is write only and the relevant bits are shown in Figure 5 - 29.

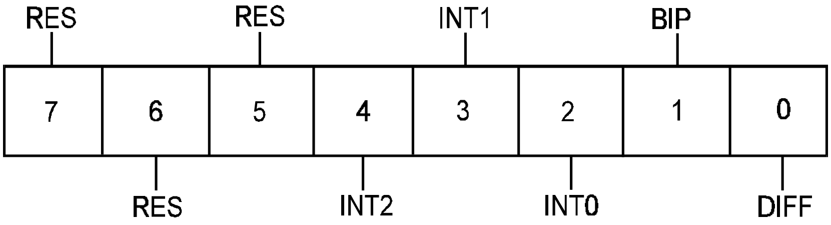


Figure 5 - 29. ADCCFG Register

The configuration of the interrupt source bits is as follows:

Bit INT2	Bit INT1	Bit INT0	Interrupt Configuration
0	0	0	No interrupt source
0	0	1	End of conversion
0	1	0	End of DMA block
0	1	1	Every counter/timer pulse
1	0	0	External Trigger*
1	0	1	External Clock*
1	1	0	Reserved
1	1	1	Reserved

Note: * External Trigger and External Clock as an Interrupt Source will only be available in the PC104-30 Rev 1C Version

EDR_SetInterruptSource programs these bits as follows:

Mode	INT1	INT0
EDR_INT_ADIN	0	1
EDR_INT_TC	1	0
EDR_INT_COUNTER	1	1

The A/D input range is configured as follows:

		Input Signal Range	
LK1	**Bit BIP**	**G, GA**	**F, FA**
1-2	0	-5V to +5V	-5V to +5V
1-2	1	0V to +10V	-10V to +10V
2-3	0	-10V to +10V	-5V to +5V
2-3	1	0V to +10V	-10V to +10V

Note that LK1 is not present on the PC104-30Fx boards. Only the BIP bit controls the A/D ranges.

EDR_SetADInRange sets or clears this bit depending on the desired range and board model.

The DIFF bit configures the A/D inputs for:

- DIFF = 1 - differential input signal.
- DIFF = 0 - single ended input signal

The DIFF bit is configured by EDR_SetADInType

DACCFG - D/A Configuration Register (Offset 29)

This register is the mode register in the D/A converter. The register is write only and the relevant bits are shown in Figure 5 - 30. EDR_SetDAOutRange configures this register.

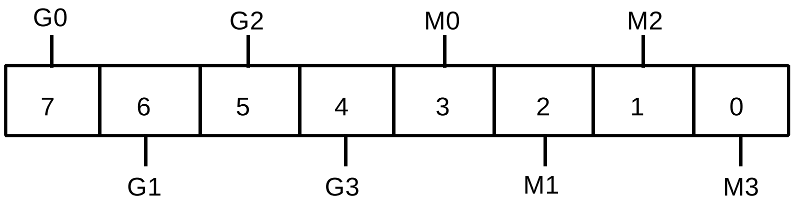


Figure 5 - 30. DACCFG Register

a. **Bits G0 to G3**

   Output gain setting bits.

b. **Bits M0 to M3**

   These bits set the output mode of each of the four DACs (DAC0 through DAC3 respectively).

   The following table shows the configuration for each combination of M and G bits.

Gx	Mx	Configuration
0	0	0V to 10V
0	1	-5V to +5V
1	0	0V to +20V
1	1	-10V to +10V

CLKSRC - Clock Source Configuration Register (Offset 30)

The register format is shown in Figure 5 - 31.

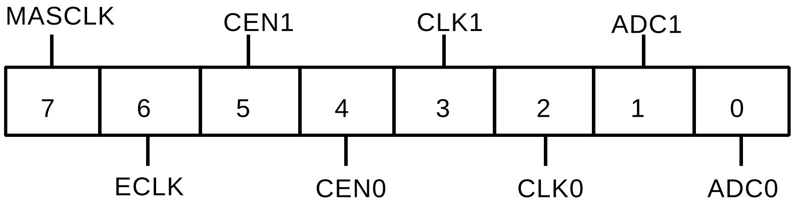


Figure 5 - 31. CLKSRC Register

a. Bits ADC 1 and ADC0

The ADC 1 and ADC0 bits define the A/D clock trigger source as shown below.

ADC 1	ADC 0	Resulting Configuration
0	0	Internal clock to A/D clock trigger and to external trigger.
0	1	External trigger disconnected, internal clock to A/D clock/trigger.
1	0	External trigger to A/D clock/trigger, internal clock disconnected.
1	1	External trigger to gate 1, internal clock to A/D clock/trigger.

Internal clock is the output from OUT1 of the counter/timer. Gate 1 is the gate of the counter/timer.

b. Bits CLK1 and CLK0

These bits configure the clock source for the counter/timer as shown in the table below.

CLK1	CLK0	Resulting Configuration
0	0	No clock source (grounded).
0	1	External clock as source.*
1	0	Clock divider output (OUT1) as clock source.
1	1	Master clock (2 or 8 MHz) as clock source.

* If bit ECLK = 0, the counter/timer clock source is disabled.

c. ECLK Bit

The ECLK bit controls the external bit as follows:

- If ECLK = 0 - external clock is output.
- If ECLK = 1 - external clock is input.

d. Bits CEN1 and CEN0

These bits control the enabling/disabling of the counter/timer 2 (Gate 2 input). The configuration of the bits is as follows:

CEN 1	CEN 0	Resulting Configuration
0	0	Always enabled (pulled high).
0	1	Enabled by external clock. *
1	0	Enabled by clock divider output (OUT1).
1	1	Always enabled (pulled high).

* ECLK must be 1.

EDR Functions that Program this Register

EDR_SetADCClock Trigger configures bits ADC1 and ADC0 as follows:

Mode	ADC1	ADC0
EDR_ADC_MASTER	0	0
ESR_ADC_INTERNAL	0	1
EDR_ADC_SLAVE	1	0
EDR_ADC_EXTTRIG	1	1

EDR_CTClockSource configures bits ECLK, CLK1 and CLK0 as follows:

Mode	CLK1	CLK0	ECLK
EDR_CS_GND	0	0	0
EDR_CS_EXT	0	1	1
EDR_CS_DIVIDER	1	0	0
EDR_CS_2MHZ	1	1	0

EDR_CTGateSource configures the ECLK, CEN1 and CEN0 bits as follows:

Mode	CEN1	CEN0	ECLK
EDR_CT_SOFTGATE	0	0	X
EDR_CT_EXTGATE	0	1	1
EDR_CS_ENABLED	1	1	X
EDR_CT_DIVOUTPUT	1	0	X

X = not changed

e. Bit MASCLK

This bit enables the user to select either the 8 MHz clock or the 2 MHz clock as the Master Clock. When this bit is not set, the clock source is 2 MHz (default). When the bit is set, the clock source is set to 8 MHz.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6: Programming Guide

Introduction

This chapter gives a "How To" guide to programming the PC104-30. To make best use of the contents of this chapter, you should be familiar with the contents of the previous chapter.

We recommend that you use our EDR Software Developers Kit to program your board instead of writing low level code. However, if you do decide to write your own low-level code, we strongly recommend that you study the source code for the old drivers. This code is found in the \PC104-30\OLD directory. When you study this code, bear in mind that the code is written to support all the old PC104-30 variants. Since these boards are all register compatible with the F and G models, you can still refer to it.

Where possible, in the following guide there are references to where you can find example code showing how to perform various functions.

Converting From Binary To Analogue Values

Analogue data from the A/D converter is always in the form of offset binary code. Analogue voltages may be calculated from the digital codes as follows:

A/D Converter Codes

a. For the 0 to + 5V range:

   Voltage = (Digital code)(5)/4096

b. For the -5 to + 5V range:

   Voltage = (Digital code - 2048)(5)/2048

c. For the -10 to +10V range:

   Voltage = (Digital code - 2048)(10)/2048

DAC0 and DAC1 Converter Codes

DAC0 and DAC1 are 12 bit converters. Note that the monopolar formulas given here assume the DACINV bit in the ADCCFG register is clear. The bipolar formulas assume it is set. This is to ensure compatibility with the old boards. Conversion is as follows:

a. For the 0 to + 10V range:

Voltage = (Digital code)(10)/4096

b. For the -10 to +10V range:

Voltage = -(Digital code - 2048)(10)/2048

c. For the -5 to +5V range:

Voltage = - (Digital code - 2048)(5)/2048

d. For the 0 to 13 V range:

Voltage = (Digital code)(20)/4096

DAC2 and DAC3 Converter Codes

DAC2 and DAC3 are 12 bit converters. Note that the monopolar formulas given here assume the DACINV bit in the ADCCFG register is clear. The bipolar formulas assume it is set. This is to ensure compatibility with the old boards. Conversion is as follows:

e. For the 0 to + 10V range:

Voltage = (Digital code)(10)/4096

f. For the -10 to +10V range:

Voltage = -(Digital code - 2048)(10)/2048

g. For the -5 to +5V range:

Voltage = - (Digital code - 2048)(5)/2048

h. For the 0 to 13 V range:

Voltage = (Digital code)(20)/4096

The above formulas all assume that the PC104-30 is calibrated as described in Chapter 7. If you do not use the recommended calibration procedure, these formulas may not apply.

Initialisation

To initialise the PC104-30, the following steps must be performed. The function Init, supplied with the old PC104-30 driver software, performs this function. This sequence must be followed prior to attempting any function. Note that, first, you should write to the three configuration registers as described in Chapter 5.

1. Write 0 to the IMUX and IGATE Registers (assuming Interrupts and DMA is not used). Note that this can only be done in DOS (NOT a DOS Box under Windows ‘95/’98/NT/2000. If you are using a Multitasking Operating System such as Window ‘95/’98/NT or 2000 then the DMA and IRQ Parameters are set in the Eagle Control Panel Appelate.
2. Write 92 (hex) to the A/D mode register (ADMDE).
3. Write 34 (hex) to the counter control register (TMRCTR). This sets the mode of the A/D clock prescaler to 2.
4. Write 74 (hex) to the counter control register (TMRCTR). This sets the mode of the A/D clock divider to 2.
5. Write B6 (hex) to the counter control register (TMRCTR). This sets the mode of the uncommitted counter/timer to 3.
6. Write 02 (hex) to the A/D control/channel register (ADCCR). This disables DMA and interrupts, and sets the A/D for software strobes.
7. Write 0 to the digital I/O control register (DIOCNTRL). This configures all digital input lines as inputs.
8. Write 0 to the GMEM0, GMEM1, GMEM2 and GMEM3 registers. This selects a channel gain of 1 for all input channels.
9. Wait at least 100 :s.
10. Read the high and low byte of the A/D data register.

The PC104-30 is then ready for operation.

Clearing the A/D Subsystem

Before using the A/D subsystem, it is important to wait for any current A/D conversion to complete, and to clear any information in the A/D data registers. The sequence below performs this function, as well as clearing the PC104-30's FIFO buffer. The function **Clean**, supplied with the old PC104-30 driver software, performs this function. This sequence must be followed prior to attempting any A/D input function.

1. Write 92 (hex) to the A/D mode register (ADMDE).
2. Write 02 (hex) to the A/D control/channel register (ADCCR). This disables DMA and interrupts, and sets the A/D for software strobes.
3. Read the high and low byte of the A/D data register.
4. Wait at least 100 :s, or until the done bit is set.
5. Read the high and low byte of the A/D data register.

Writing to the D/A Converters

To write to the D/A converters, you must convert the required voltage to a digital code, and then write this to the appropriate registers. Remember that, in the case of the 12-bit converters, the code must be shifted left by 4 bits, and that the D/A output is not updated until the low byte register is written.

Digital I/O

Digital I/O is performed by reading or writing the required digital values to the appropriate registers. You must however remember to configure the port in use before reading and writing. The initialisation procedure above configures all the digital ports as inputs.

Setting Channel Gain

The gains of channels 0, 4, 8 and 12 are set in the GMEM0 register, the gains of channel 1, 5, 9 and 13 in the GMEM1 register, the gains of channel 2, 6, 10 and 14 in the GMEM2 register and the gains of channel 3, 7, 11 and 15 in the GMEM3 register. For example, to set the gain of channels 0, 1 and 4 to 1000, and that of all other channels to 1 the following values would be used:

GMEM0	**00001111 (binary)**
GMEM1	**00000011 (binary)**
GMEM2	**00000000 (binary)**
GMEM3	**00000000 (binary)**

Obtaining a Single A/D Reading

To obtain a single A/D reading under program control, proceed as follows:

1. Clear the A/D subsystem as previously described.
2. Set the channel gain as previously described.
3. Write a byte containing the address of the channel you wish to convert, with the STBC bit set, and all other bits cleared, to the ADCCR.
4. Write the same byte, but with the SSTB bit set as well as the STBC bit, to the ADCCR.
5. Write the same byte, but with the SSTB bit cleared, to the ADCCR.
6. Wait for the Done bit in the ADDSR to be set.
7. Read the result from the ADDSR and the ADDATL registers.

Setting the Sample Rate

Assuming that the PC104-30 is initialised as previously described, setting the sampling rate is simple:

1. Decide on values for the A/D clock prescaler and divider. For example, to sample at l00 kHz, you could set the prescaler to 2, and the divider to 10 (or vice-versa). Remember that the maximum value for either the prescaler or the divider is FFFF(hex).
2. Write the LSB of the prescaler value to the PRESCALER register, then the MSB.
3. Write the LSB of the divider value to the DIVIDER register, then the MSB.

The sampling rate is then set. Note that if the board is not jumpered for internal clock, then neither of the above registers will have any effect.

The procedures Ad_prescaler and Ad_clock (in the file PC104-30S.C) show how to perform the above functions.

Loading the Channel List/Block Counter

The easiest way to perform multi-channel data input operations on the PC104-30, is to make use of the channel list. Any sequence of channels, in any order, up to a maximum length of 31, can be loaded into the channel list. On completion of each A/D conversion, the PC104-30 automatically loads the next channel into the channel address register. The procedure for loading this channel list is as follows:

1. If you are using block mode, then write (257 - N) to the BLKCNT register, where N is the number of samples per block.
2. Clear the A/D subsystem as previously described.
3. Write the first channel address to the ADCCR. All other bits except the STBC should be cleared. The channel list now contains only the first channel.
4. Set the A/D mode register (ADMDE) to 9F(hex). This sets the channel list mode to add.
5. Write, in sequence, the rest of the channels to be converted to the ADCCR, as above. Note that on read the ADCCR will reflect the first channel address written.
6. Set the A/D mode register (ADMDE) to either 90 (hex) (for normal trigger mode) or 91 (hex) (for block trigger mode). This sets the channel list mode to ignore. You can now write control bits to the ADCCR without disturbing the channel list.

The channel list is now ready for operation.

The driver procedure Ch_list_load, found in the file PC104-30S.C, shows how the driver software performs this function.

If you need to convert only a single channel, all that is required is to write the channel address to the ADCCR. It's still a good idea to set the mode register to 90 (hex), as previously described, as this enables the PC104-30 FIFO.

Obtaining a Series of A/D Conversions by Polled I/O

Polled I/O is by far the simplest way to obtain a sequence of samples. It is however limited to about 5OkHz on PC104-30 boards. The procedure is as follows:

1. Set the sampling rate, as previously described.
2. Set the channel gain, as previously described.
3. Load the channel or channels to be converted into the channel list, and set the block counter as previously described.
4. Set the STBC bit in the ADCCR to 0. This enables A/D strobes.
5. Wait for the A/D done bit in the ADDSR to be set. As soon as it is, read the A/D result into memory.
6. Repeat step 4 until you have collected as many samples as you require.
7. When the sampling procedure is complete, set the STBC bit to 1.

The procedures S_chan (for single channel operation) and Mb_chan (for multi-channel operations) in the file PC104-30S.C show how the old driver software performs this function.

Interrupts

Interrupt based I/O allows the PC's CPU to perform other tasks while the PC104-30 acquires data. It is however limited to low speed applications. Throughput of about 10 kHz is typical. Note that if you intend to write your own interrupt based routines, you must have a thorough understanding of both the PC and the operating system in use. A complete description of interrupt handlers is beyond the scope of this manual. However, the basic procedure is described below:

1. Configure the IRQ Channel you are going to use using the IMUX and IGATE Registers
2. Set the sampling rate, as previously described.
3. Set the channel gain, as previously described.
4. Load the channel or channels to be converted into the channel list, and set the block counter as previously described.
5. Set the PC's interrupt vector to the address of your interrupt handling procedure. This procedure must read in the results of the A/D conversion, as well as halt operations when sufficient samples have been obtained. Remember also that the interrupt handler must send an EOI (end of interrupt) command to the interrupt controller. In the case of a PC104-30PG using an

interrupt level greater than 7, both interrupt controllers must receive this command.

6. Set the interrupt enable bit in the ADCCR to 1. This enables the PC104-30 interrupts.
7. Set the STBC bit in the ADCCR to 0. This enables A/D strobes. The program can then continue with other work.
8. As soon as an A/D conversion completes, control is passed to the interrupt handling procedure. This continues until the interrupt handler disables interrupts.
9. When the sampling procedure is complete, set the STBC bit to 1, and the interrupt enable bit to 0.

The procedures Mi_chan, Int_chk and Int_close in the file PC104-30S.C show how the driver software performs this function.

Single Channel DMA

Note that the "single channel" refers to the number of DMA channels used, NOT to the number of input channels that can be sampled. All the PC104-30 boards described in this manual can scan multiple channels under the control of the channel list hardware. DMA is normally the only way to achieve full throughput on the various PC104-30 boards, and also allows the program to continue with other activities while the DMA takes place.

If you intend to write your own DMA based routines, you must have a thorough understanding of both the PC, and the operating system in use. A complete description of DMA procedures is beyond the scope of this manual. The basic procedure is described below:

1. Configure the Primary and Secondary DRQ and DACK Channels using the IMUX Register [Offset 18] and the IGATE Register [Offset 19].
2. Set the sampling rate, as previously described.
3. Set the channel gain, as previously described.
4. Load the channels to be converted into the channel list, and set the block counter as previously described.
5. Set up the PC's DMA hardware with the address of the section of memory into which you wish to transfer the results of the A/D conversions. Also remember to enable the DMA level you intend to use. The DMA controller should be programmed for demand mode operation.
6. Set the DMA enable bit in the ADCCR to 1, and the DMA mode bit in the ADMDE register to 1. This enables the PC104-30 DMA. In the case of the

PC-30PG, it also sets the DMA mode to terminate on TC. This is not strictly necessary, but should be done for compatibility with Micro channel products.

7. Set the STBC bit in the ADCCR to 0. This enables A/D strobes. The program can then continue with other work.
8. As soon as an A/D conversion completes, the results of the conversion are transferred to memory. This continues until the count value programmed into the DMA hardware in the PC reaches 0.
9. When the sampling procedure is complete, set the STBC bit to 1, the DMA mode bit to 0, and the DMA enable bit to 0.

Note that the memory space to which the data is transferred should start on an even byte, and that the entire memory space must be inside a single 128K physical page.

The procedures Sd_chan (single channel operation), Mbd_chan (multi-channel operations), Dma_chk and Dma_close in the file PC104-30C.C show how the old driver software performs this function.

Dual Channel Gap-free DMA

The PC104-30 can perform dual channel gap-free DMA. Normal DMA can only transfer up to 65536 samples, but dual channel DMA can transfer to the limits of installed memory. Dual channel DMA also has the advantage the restrictions due to the host PC's requirement that the entire memory space must be in a single 128K block fall away.

Dual channel gap-free DMA is very similar to conventional DMA. The basic procedure is as follows:

1. Configure the Primary and Secondary DRQ and DACK Channels using the IMUX Register [Offset 18] and the IGATE Register [Offset 19].
2. Set the sampling rate, as previously described.
3. Set the channel gain, as previously described.
4. Load the channel or channels to be converted into the channel list, and set the block counter as previously described.
5. Split the memory space into which you wish to transfer the data into segments. Each segment must be chosen so that only the lower 17 bits of the physical address vary from start to end.
6. Set up the primary DMA channel with the address of the first segment, and the secondary DMA channel with the address of the second segment. Enable both levels. The DMA controller should be programmed for demand mode operation.

7. Set the DMA enable bit in the ADCCR to 0, and the DMA mode bit in the ADMDE register to 1. This enables the PC104-30 DMA in the swap on TC mode.
8. Set the STBC bit in the ADCCR to 0. This enables A/D strobes. The program can then continue with other work.
9. As soon as an A/D conversion completes, the results of the conversion are transferred to memory. This continues until the count value programmed into the DMA hardware in the PC reaches 0.
10. Once the primary DMA channel completes, the PC104-30 automatically swaps to the secondary channel, and sets the DMA enable bit in the ADCCR. You must now reprogram the primary DMA channel with the address of the next segment, and clear the DMA enable bit.
11. When the secondary DMA channel completes, the PC104-30 swaps back to the primary DMA channel, and again sets the DMA enable bit in the ADCCR. You must then reprogram the secondary DMA channel with the address of the next DMA channel, and clear the DMA enable bit.
12. This process of swapping buffers continues until the last segment in the memory space. When this occurs, rather than clearing the DMA enable bit, you should leave it set. The PC104-30 then automatically shuts down DMA operation at the end of the last segment.
13. When the sampling procedure is complete, set the STBC bit to 1, the DMA mode bit to 0, and the DMA enable bit to 0.

The procedures Mdh_chan, Dma_chk and Dma_close in the file PC104-30C.C show how the old driver software performs this function.

DMA Data Format

After DMA has completed, the format of the data in the DMA buffer is as follows:

	Bit 7 (MSB)				Bit 0 (LSB)
Byte 0	Sample 0 Bits 7 - 0				
Byte 1	-		-	Trig	Sample 0 Bits 11 - 8
Byte 2	Sample 1 Bits 7 - 0				
Byte 3	-		-	Trig	Sample 1 Bits 11 - 8
Byte 4	Sample 2 Bits 7 - 0				
Byte 5	-		-	Trig	Sample 3 Bits 11 - 8
	• • • •				

Dealing With Extended Memory

Any version of the PC104-30 can execute DMA transfers into extended (NOT expanded) memory. This is done precisely as described above. Note however that you will need to make use of BIOS functions to obtain this data, as your program cannot access this memory directly under DOS.

The procedures Mde_chan, E_mem_size and Xfer_dma_res in the file PC104-30C.C show how the old driver software accesses extended memory.

Error Detection

All versions of the PC104-30 described in this manual implement error detection. If an error occurs, the error bit in the ADDSR is set. There are two approaches to checking this bit:

a. Check only at the end of an operation. As long as you do not write a one to the error reset bit in the ADMDE, the error bit will remain set. You need hence only check the bit at the end of an entire operation. Note however that you must check the bit fast enough so that it is not set due to A/D conversions which occur after you have obtained all the required samples. For this reason, this technique is not suitable for error detection of DMA operations, as the error bit will almost inevitably have been set before a program can check it, regardless of whether it was or was not set during the actual DMA transfer.

b. Check the stored data. The PC104-30 has been designed in such a way that the error detection bit is in the same register as the A/D data. It can hence be transferred to memory along with the data, at no extra overhead. This always occurs during DMA transfers. Error checking can thus be performed on the actual data, after the completion of sampling.

Almost all of the routines in the old PC104-30 driver software implement error detection.

End of DMA Block Interrupt

Note that for the PC104-30 to generate an end of DMA block interrupt, the DMA mode must be set to "swap on TC, as discussed in the description of the ADMDE register. The interrupt source must also be set to “End of DMA block” as discussed in the description of the ADCCFG register.

Chapter 7: Calibration

Introduction

This chapter contains information on the calibration procedures for the A/D and D/A sub-systems on the PC104-30 series of boards.

These procedures should be performed at six month intervals, or whenever the input or output range jumpers are changed.

Allow the host PC and the board to warm up for at least one hour before calibration.

A/D Calibration

A/D calibration is performed by adjusting three trimpots. These trimpots are easily located from the board layout shown in Appendix C, or the labels on the PC104-30 board itself.

Requirements

a. Calibration is done on channel 1. The recommended connector wiring is shown in Figure 7 - 1.

b. Calibration is performed with the board configured into its intended operating mode.

c. All cables should be as short as possible. Note that screened cable is preferable in order to minimise noise interference.

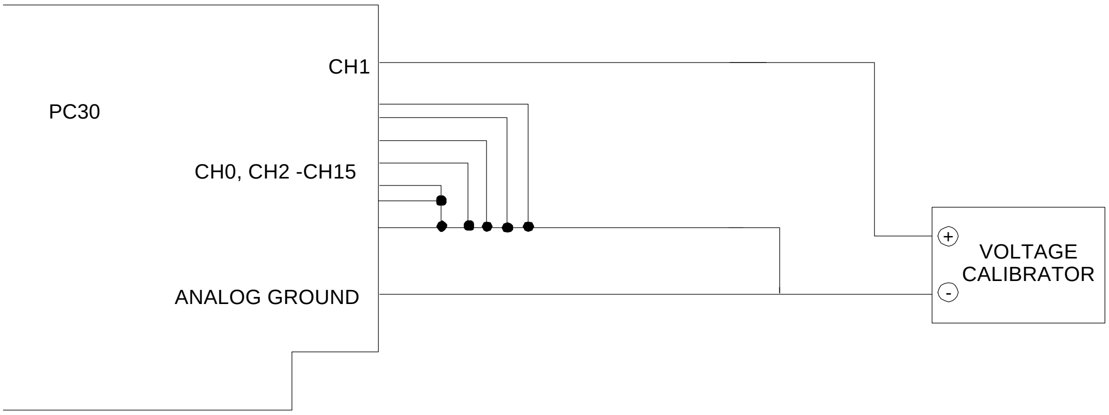


Figure 7 - 1. A/D Calibration Connections

Equipment Required

a. Precision voltage source. Range +10V to -10V, with an absolute accuracy better than 0.005%, resolution 100 nV or better.

b. Precision digital multimeter with ±10V range, absolute accuracy better than 0.0005%, resolution 100 nV or better.

Setting the Reference Voltage

Before calibrating the A/D, the reference voltages must be set properly. This is normally done during manufacturing, but if the voltage does not match, re-calibrate as follows:

+10V

Connect the multimeter to analogue ground and U16 Pin 1 or C24 +ve Side (Indicated on the component by a line). The voltage reading should be +10.0000V. If the voltage is out of specification, adjust VR4.

Another method is to set DACs to ±10V output. Set DAC output to +10V using Waveview. Connect SENSE0 [Pin 19 of AD-CON Connector] to DAC0 [Pin 20 of AD-CON Connector]. Measure the output voltage on the output of DAC0 (ie: Pin 19 of IDC26 Male Header [AD-CON]). Adjust VR4 until a voltage of 9.995V is reached.

Calibration Procedure

Run the CAL30FG.exe program installed in the PC104-30 directory.

A/D Calibration for the PC104-30G/PC104-30GA Boards

Bipolar Mode

1. Adjust A/D for maximum gain (ie. 1000) and apply 0.00mV to Channel 1 (all other channels must be grounded to AGND). Adjust VR1 (PC104-30FG Rev1B), the instrumentation amplifier offset pot, for output 800H.
2. Set A/D for a gain of 1 and apply (-FS+2LSB) to channel 1 (ie. -4.9988V for ±5V or -9.9976V for ±10V). Adjust VR2 (PC104-30FG Rev1c), bipolar A/D offset, for an output code which flickers between 000H and 001H.
3. Set A/D for a gain of 1 and apply (+FS-3/2LSB) to channel 1 (ie. +4.9963V for ±5V range or +9.9927V for ±10V). Adjust VR5 (PC104-30FG Rev1c),gain pot, for an output code which flickers between FFEH and FFFH.
4. Repeat the above steps until no further adjustment is required.

Monopolar Mode

1. Adjust A/D for maximum gain (ie. 1000) and apply 0.00mV to Channel 1 (all other channels must be grounded to AGND). Adjust VR1 (PC104-30FG Rev1B), the instrumentation amplifier offset pot, for output code 000H.
2. Set A/D for a gain of 1 and apply (FS+½LSB) to channel 1 (ie. for 0 to 10V range it should be +1.22mV). Adjust VR3 (PC104-30FG Rev1B), A/D offset, for an output code which flickers between 000H and 001H.
3. Set A/D for a gain of 1 and apply (+FS-3/2LSB) to channel 1 (ie. +9.9963V). Adjust VR6 (PC104-30FG Rev1B),gain pot, for an output code which flickers between FFEH and FFFH.
4. Repeat the above steps until no further adjustment is required.

A/D Calibration for the PC104-30F/PC104-30FA Boards

Bipolar Mode (±5V)

1. Adjust A/D for maximum gain (ie. 1000) and apply 0.00mV to Channel 1 (all other channels must be grounded to AGND). Adjust VR1 (PC104-30FG Rev1B), the instrumentation amplifier offset pot, for output code 800H.
2. Set A/D for a gain of 1 and apply (+FS-3/2LSB) to channel 1 (ie. +4.9963V). Adjust VR7 (PC104-30FG Rev1B), gain pot, for an output code which flickers between FFEH and FFFH.
3. Set A/D for a gain of 1 and apply (-FS+½LSB) to channel 1 (ie. -4.9988V). Adjust VR3 (PC104-30FG Rev1B), bipolar A/D offset, for an output code which flickers between 000H and 001H.

4. Repeat the above steps until no further adjustment is required.

Note that steps 2 and 3 above are inter-related, and it therefore requires some expertise to enable ± Full Scale Convergence.

Bipolar Mode (±10V)

1. Adjust A/D for maximum gain (ie. 1000) and apply 0.00mV to Channel 1 (all other channels must be grounded to AGND). Adjust VR1 (PC104-30FG Rev1B), the instrumentation amplifier offset pot, for output code 800H.
2. Set A/D for a gain of 1 and apply (+FS-3/2 LSB) to channel 1 (ie. +9.9927V). Adjust VR8 (PC104-30FG Rev1B), gain pot, for an output code which flickers between FFEH and FFFH.
3. Set A/D for a gain of 1 and apply (-FS+½LSB) to channel 1 (ie. -9.9976V). Adjust VR9 (PC104-30FG Rev1B), bipolar A/D offset, for an output code which flickers between 000H and 001H.
4. Repeat the above steps until no further adjustment is required.

Note that steps 2 and 3 above are inter-related, and it therefore requires some expertise to enable ± Full Scale Convergence.

A/D Calibration Software

The program CAL30FG.EXE, supplied on the distribution disk, automates the above procedure. Note that for correct operation, the set-up information supplied in the first menu must be correct.

DAC0 to DAC3 Calibration

In general, you do not need to calibrate the DACs at all, because it has already been done when adjusting the reference voltages (ie. usually during manufacturing). However, if you require a different full scale (+10 down to +8V) output on these DACs, proceed as follows:

1. Run the CAL30FG.exe program from the EDR sub-directory.
2. Select DAC calibration. Alternatively, you can run Waveview and set DAC0 (with SENSE0 shorted to DAC0) to maximum full scale (ie. 10V in WaveView).
3. Connect the multimeter to the output of DAC0 (with SENSE0 shorted to DAC0) and analogue ground.
4. Adjust the reference pot VR4 (PC104-30FG Rev 1B) until you obtain the required voltage output.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8: Appendix A: Hardware Specifications

Table A - 1: Analogue Inputs	
Number of Channels Number of Channels with simultaneous sample/hold	16 single-ended or 8 differential (software selectable) 16 single ended only
Resolution	12-bits (1 in 4096)
Total System Accuracy (absolute accuracy)	±1 bit LSB (for Gain of 1) Note: Lab Tested @ room temp
Linearity: Integral Differential	±0.05% FS ±:¾ LSB max.
A/D Input Voltage - Ranges	±5V, ±10V, 0 to 10V (PC104-30G, PC104-30GA) ±5V, ±10V (PC104-30F, PC104-30FA)
Data Acquisition Rate	PC104-30Gx: 100kHz, PC104-30Fx: 330kHz (G<1000) PC104-30Gx: 100kHz, PC104-30Fx: 100kHz (G=1000)
Input Impedance:	
On Channel	10M/20pF
Off Channel	10M/100pF
Offset Voltage	±5 LSB adjustable to 0
Input Bias Current	±100 pA/°C
Input Bias Offset Drift	±30ppm/°C
Input Gains:	
Ranges	1, 10, 100, 1000 or 1, 2, 4, 8 (S/W selectable)
Gain Error	Adjustable to 0
Gain Accuracy	0.25% max, 0.05% typical for gains < 1000
CMRR for various gains	1% max, 0.1% for g=1000
Monotonicity	0 to 70°C
Temperature Drift:	
Full Scale Error Drift	6 ppm/°C (PC104-30Fx)
Bipolar Zero Drift	1 ppm/°C (PC104-30Fx)
Gain	±30 ppm/°C

Table A - 1: Analogue Inputs	
Input Over voltage Protection	±35V (With Power ON), ±25V (Without Power)
A/D FIFO Buffer Size	16 samples
Channel Gain/Queue Length	31
A/D Clock: Internal Clock Clock frequency tolerance Clock Drift Internal Clock Divider External Clock External Trigger Channel List (queue) Length Block Scan Mode	2 MHz or 8 MHz (software selectable) 0.01% 10 ppm/°C 2 x 16 bit stages TTL compatible TTL compatible 31 Up to 256 channels per block; all channels converted at max. thruput on each clock pulse
Noise Levels (p-p)	G=1: ±1 bit; G=10: ±1 bit; G=100: ±2 bits Noise levels will vary according to environmental conditions
EOB (Effective No of Bits)	G=1: ±1 bit; G=10: ±1 bit; G=100: ±2 bits Noise levels will vary according to environmental conditions
Data Acquisition Modes	Polled I/O, Interrupts, Single and Dual Channel DMA

Table A - 2: Analogue Outputs	
Number of channels	4
Resolution	Four 12-bit
Accuracy	±1 LSB (12-bit)
Differential Nonlinearity	±1 LSB max.
Output Ranges	±5V, ±10V, 0 to 13 V (software selectable)
Offset Error	Unipolar: 3 LSB typical, 1 LSB max. (12 bit) Bipolar: 2 LSB typical, 2 LSB max. (12 bit)
Gain: Ranges Error	x1, x2 2 LSB typical, 5 LSB (12 bit)
Settling time to 2 LSB	10 µs max. in a Load of 500 p, 2 kΩ
Throughput Rate	500 kHz (depending on computer)

Table A - 2: Analogue Outputs	
Temperature Drift	100 ppm/EC of full scale
Max. Current Output Source	±5 mA maximum
Monotonicity	0 to 70°C

Table A - 3: Digital I/O	
Number of I/O Lines	24 in 3 ports (8255 PPI)
Voltage Compatibility	TTL
Interface Selection	Programmable for simple I/O, strobed I/O or handshake I/O
Max. Input Voltage	5.5V
Max. Current Source/Sink	±2 mA

Table A - 4: Timer/Counter Specifications	
Resolution	16 bits
Voltage Compatibility	TTL
Number of Counters	3 (2 used for A/D timing)

Table A - 5: PC Interface	
Base Address	0 - 1FFF DIP Switch selectable
Number of Registers	Thirty Two 8 bit registers
Interrupts	Register selectable for End of Conversion, DMA block, Timer, External Trigger* or External Clock* * Only available in the PC104-30 REV1C Version
DMA	Single/Dual channel software selectable to levels 5, 6 or 7
Analog I/O Connector	2.00mm IDC 26 Male Header
Digital I/O Connector	2.00mm IDC 40 Male Header

Table A - 6: Environmental Specifications	
Operating Temperature	0 to 70°C (Commercial Vers) -40°C to 85°C (Industrial Vers)

Storage Temperature	-55 to 150°C
Relative Humidity	5% to 95% non-condensing

Table A - 7: Power Requirements	
+5 V	750 mA typ.

Table A - 8: Physical	
Dimensions	90.2mm x 95mm (Standard PC104 Dimensions)

Table A - 9: Software Support
Register compatible with PC30F/G series boards
Supported by EDR Software Development Kit
DOS language support
Windows Win '95/'98/'NT4 language support
Win 2000 language support available in Dec '99
Labview, LabWindows/Labtech Notebook drivers available
Visual Basic Custom Controls available
DASYLab support
Test Point support
Matlab support

Chapter 9: App B: Compatibility

Introduction

This appendix discusses the compatibility of the PC104-30F and G models with the other series PC30 boards (PC-30B, PC-30D, PC-30DS and PC-30PG) and with the even older series boards (PC26, PC-30 and PC39).

New Series Boards

The PC104-30F and G boards are completely compatible with all the new series boards PC-30G/GA/F/FA boards with the exception of the IRQ/DMA configuration registers.

Running Old Software on the PC-30F and G

Pre PC-30F and G software can be run on the PC104-30F and G as long as the configuration registers have been set to a known state prior to running the old software. Included on the distribution disk is a utility program which sets the registers to emulate jumper settings on the old boards. This is INIT104.EXE, and is used as follows:

INIT104 [/c] [<configfile>]

The <config file> parameter is the name of the configuration file previously created with this utility. If you do not specify a config file, INIT104 will allow you to create a new one. If the /c switch is used, then the board is configured as per the config file and the program immediately returns to the command prompt. This is useful in batch files.

Running PC104-30F and G Software on Older "New Series" Boards

Any PC104-30F and G software can be run on the PC104-30B, PC104-30C, PC104-30D, PC104-30DS and PC104-30PG without modification. Accesses to the configuration registers are ignored. The jumpers on the board **must** be configured correctly.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 10: Appendix C: Layout Diagram

A typical component layout for the PC104-30F and G boards is shown on the following pages.

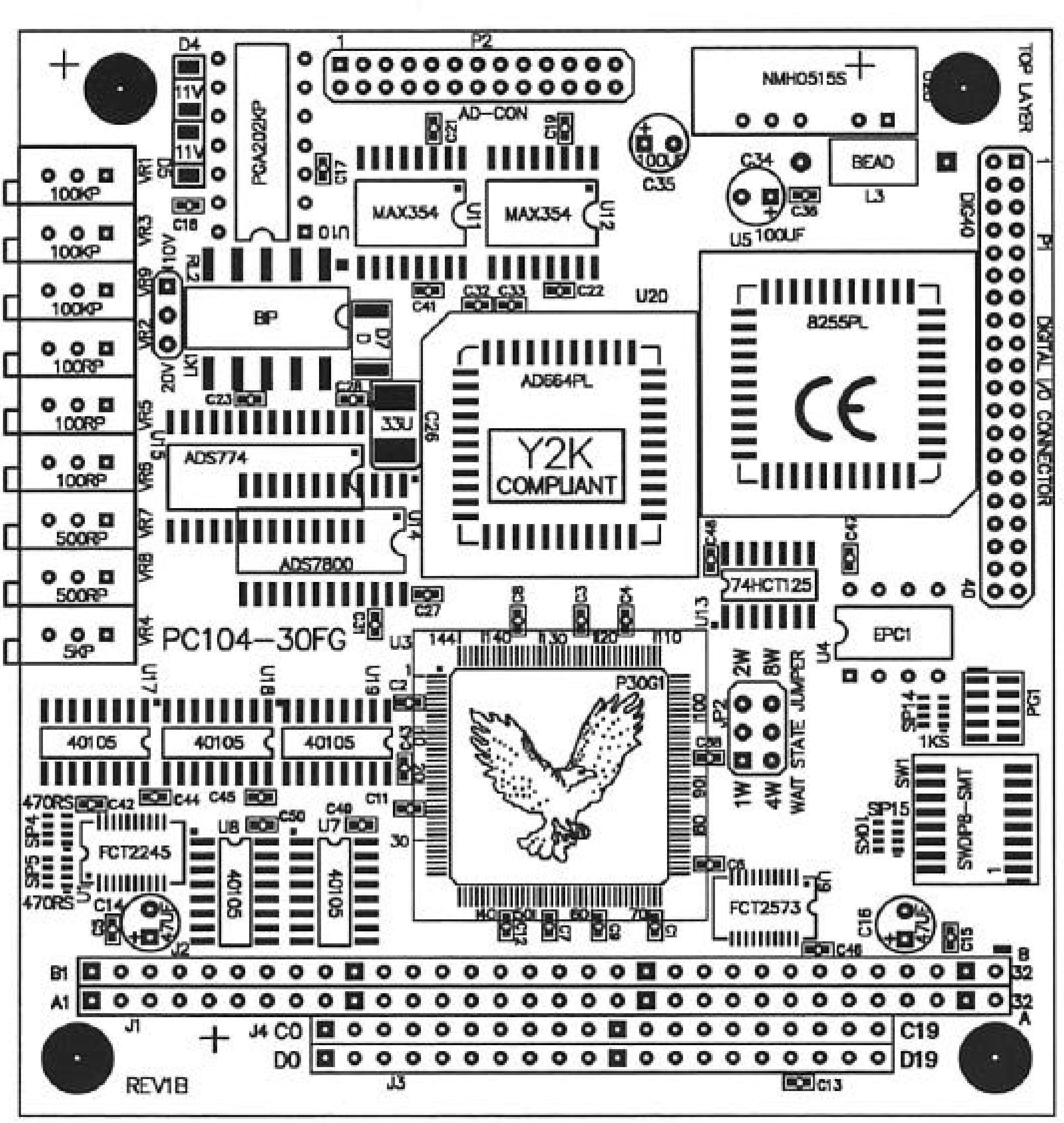


Figure B - 32. PC104-30F/G Template

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 11: Appendix D: Problem Determination Guide

Introduction

If you are experiencing problems, first check the following:

1. Remove the PC104-30, and check that all ICs are firmly seated in their sockets, that there is no obvious damage to any components, and that the edge connector fingers on the PC104-30 are clean.
2. Check that the PC104-30 is jumpered correctly for your application.
3. Replace the PC104-30, and check that it seats firmly in the PC104 Connectors. Also check that no components are touching an adjacent board.
4. Check that the cables are securely plugged into the PC104-30.

The Diagnostics Function

The PC104-30 contains a very comprehensive diagnostics program. All of the supplied demo programs, Waveview, as well as the calibration program use this, and it can be used to diagnose malfunctions on the PC104-30. In fact, the only PC104-30 malfunctions which it will not detect are the following:

a. Damaged input multiplexer.

b. Damaged D/A output amplifier.

c. Damaged digital input or output lines.

Common Problems

Waveview Cannot Find the Board

This is typically as a result of incorrect jumper settings or a damaged board.

A/D Output Code All Zeros or All Ones

This is typically as a result of floating inputs, or an overload.

It is possible that the Multiplexer is faulty probably because the max voltage specification was exceedcd. Although the PC104-30 has Analog Input protection of ±35V (Power ON) and ±25V (Powered Off), it is possible that a spike > the above spec entered the analog input channels thereby destroying the Mux. Return the Board to your nearest distributor for repairs.

A/D Readings Are Noisy

This can be caused by one of the following:

a. Long leads.

b. An electrically noisy environment.

c. Overloads on other input channels. Note also that if an input channels is overloaded it may saturate in such a way as to give a reading which appears to be in the normal range, but is very noisy.

d. Excessive source resistance. The source resistance of the devices connected to the inputs of the PC104-30 should not be greater than l K.

First Reading in a Series is Inaccurate

This is normally as a result of an overload on another input, or long leads, or a very high source impedance.

The Board Does Not Operate at Full Throughput

The PC104-30 is a very high performance board, and makes correspondingly high demands on the PC in which it is installed. If you find that your PC104-30 cannot operate at full throughput, here are some points to check.

a. To achieve full throughput, you will need to make use of DMA transfers. Program transfer techniques cannot transfer data sufficiently fast to achieve full throughput on the PC104-30.

b. If your PC has a high speed or "Turbo" mode, make sure the PC is in this mode. You may have to run a special program, or press a combination of keys on the keyboard to switch to high speed mode. Consult the manual supplied with your PC for more information. Note that the speed setting in most clones effects not only the CPU speed, but the DMA speed as well.

c. Some PC compatibles have a jumper setting for number of bus wait states. This should be set to the minimum. Consult the manual supplied with your PC for more information.

Board Does Not Operate in Block Trigger Mode

When using block trigger mode, you must remember that the board operates at its maximum throughput; in fact, generally significantly faster than the rated maximum. Hence you must use DMA for block mode transfers, except in the case of a block length no greater than 16. The PC104-30 can store up to 16 samples in its FIFO buffer, so easing the transfer rate requirements on the host PC.

'Board not Found' Message Appears when Running Test Software (ie. DEMO Programs, Waveview etc.)

Another I/O card might be using the same base address location as the PC104-30F/G. Try a different base address other than the manufacturers default base address (eg. A00H) and re-run the test software. If it still persists, then try a different computer. If the test still fails, then the PC104-30F/G board is faulty. Return the board to your distributor for repairs.

CAUTION

Parts replacement on the Board must only be done by a suitably TRAINED ENGINEER OR TECHNICIAN with the appropriate tools. Failing this, return the board to your distributors for repairs.

Readings from Channel 0 through 15 are Erratic and Unstable

It is possible that the multiplexers on the PC104-30F/G or the instrumentation amplifier is overloaded or damaged. First check the operation of the board using Waveview. Check the set-up to see if the board was detected. If it was detected, press <F8>. The voltages should be displayed on the screen. If it is still erratic, then replace U11, U12 first and rerun Tests. If the problem persists then replace U10 (socketed) and re-run Waveview. The problem should be solved.

Sampled Data Via Single or Dual Channel DMA are Erroneous

1. Run WaveView in its default configuration.
2. Feed in some voltages in channel 0 to 3 and ground all unused channels.
3. Select a sampling rate of 25 kHz (ie. 4 channels gives 100 kHz) and enable channels 0, 1, 2, 3.
4. Press <F7> (ie. sampling via DMA) and check the voltages displayed. If the data is totally incorrect, try the following:
   a. It is possible that the BUS CLK on your computer is too high and hence does not conform to the PC104 Standard Spec. The PC104-30F/G was designed to run on 10% compatible PC104 Systems.
   b. Try setting the BUS clock to about 8 MHz or less. This is normally done in the **CMOS setup: Extended Chip Setup**.
   c. In the Award BIOS Setup, BUS CLK normally defaults to 12 MHz (ie. PCLK/8). Disable the Autoconfig function and change the setting to 7.195 MHz (or PCLK/12) by pressing the <PgDn> key as required. Also, increase the wait state settings in the ISA Write Cycle and the 16 bit ISA I/O Command. These are normally set to 0WS. Increase it to the maximum by pressing <PgDn>. Press <Esc>, save the CMOS setup, then reboot the computer.
   d. In the AMI BIOS, the BUS CLK normally equals PCLK/x, where x is the divisor. Disable the Autoconfig function and change x to the maximum value by pressing the <PgDn> key as required. If the wait state settings for ISA Write Cycle and the 16 bit ISA I/O Command are available, then increase it to the maximum by pressing <PgDn>. Press <Esc>, save the CMOS setup , then reboot the computer.

      The recommended settings are:

      i. For the PC104-30G/GA/GAS boards, you should set the BUS CLK and Write Cycle Wait States as follows:

         BUS CLK: 7.195 MHz

ISA Write Cycle:	1 WS
16 bit ISA I/O Command:	0 WS

ii. For the PC104-30FX boards, you should set the extended CMOS chipset configuration as follows:

BUS CLK:	7.195 MHz (or 8 MHz)
ISA Write Cycle:	0 WS
16 bit ISA I/O Command:	0 WS

5. Run WaveView again using the configuration listed above. Press <F8> and check the data on the screen. The data should be correct this time.

Note that if you cannot solve the problem, then call your distributor for immediate assistance.

DMA and Polled I/O Voltage Mismatch

In certain clone PC104 Systems, the voltages displayed using polled I/O (or Voltmeter function in Waveview) might be slightly offset by about 2mV to 4mV than when using DMA (normal/burst mode). This is normal since the calibration is done only in burst mode. The reason for the offset is that in DMA mode, the PC104-30F/G circuitry is more active. If the PC104 Power Supply is not stable it might cause a shift in the AGND level by about 2mV to 4mV.

To solve this problem (if it occurs), you should calibrate your board in one mode only.

Sampled Data Indicates Excessive Noise Levels

The probable causes of this fault are:

1. **Improper Connections to the Channels, Analogue Ground**

   Ensure that each analogue channel is properly shielded and its return line connected to Analogue Ground (ie. at the IDC26 Male Header Pin 19). Military grade shielded cable can keep noise reduction to a minimum.

   Your Load Input Impedance is probably > 1kΩ. You can also decrease the input impedance by connecting a resistor from the input channel to AGND (eg. 100kΩ resistor from CH1 to AGND). Some experimentation will be required to determine the most suitable resistor.

   Note that you should only decrease the input impedance if the noise levels exceed 20 mV.

2. **One of the Analog Input Channel gives erratic readings.**

It is possible that the Multiplexer is faulty probably because the max voltage specification was exceedcd. Although the PC104-30 has Analog Input protection of ±35V (Power ON) and ±25V (Powered Off), it is possible that a spike > the above spec entered the analog input channels thereby destroying the Mux. Return the Board to your nearest distributor for repairs.

3. **Excessive Environmental Conditions Appear to Induce Noise on the Analogue Input Lines**

   If this problem occurs, it is very difficult to identify the exact noise source origin. The best solution is to use digital averaging to filter the noise.

   Since noise is often cyclic over a time period, the easiest way to filter it is to use digital averaging. For example, assuming you are sampling two channels at 10kHz in burst mode, try oversampling it five times and take five readings. In other words, set the burst frequency to 100kHz and take five samples for each channel. Next, add each of the same five channels voltages together and divide by five. This will give you an accurate and stable result.

   This will be the easiest way to filter out the noise in extremely harsh environments.

DAC Outputs clips to ±13V irrespective of DAC Input Data

The probable causes of this fault is:

1. **DAC SENSE Lines are not connected to the appropriate DACS.**

   If the DAC Sense Lines are not used, you MUST short the SENSE Lines to the appropriate DACS. Table below lists the DACs and SENSE Lines together with their pin numbers.

DAC Number	DAC Pin	SENSE Lines	SENSE Pin Number
DAC0		0	0
DAC1		0	0
DAC2		1	1
DAC3		1	1

If the SENSE Lines are connected to the appropriate DAC and the DAC Output Voltages does not reflect the input data written then the DAC might be faulty. Replace U20 (socketed) or return the Board to your distributor. If and only if you are a Professional Engineer/Technician, you may replace the U20 yourself. Contact your distributor for a replace DAC [AD664JP] or contact your nearest Analog Devices Distributor.

CAUTION !

Parts replacement on the Board must only be done by a suitably TRAINED ENGINEER OR TECHNICIAN with the appropriate tools. Failing this, return the board to your distributors for repairs.

Digital I/O lines does not read the correct values on the Ports when configured as Inputs.

The probable causes of this fault is:

1. **I/O ports not are configured as Inputs or IC is faulty.**

   Use EDR to configure the I/O Ports correctly. If EDR is used and the problem persists then the PPI is faulty probably because the max input voltage specification was exceedcd. Ensure that the Inputs are TTL Compatible (Min Voltage: 0V, Max Voltage: 5.2V). Replace U5 (socketed) or return the Board to your distributor. If and only if you are a Professional Engineer/Technician, you may replace the U5 yourself. Contact your distributor for a replacement NEC 71055L or Intel I82C55-10 [All PLCC44 Version] or contact your nearest Intel, NEC, UMC or Tundra Distributor.

CAUTION

Parts replacement on the Board must only be done by a suitably TRAINED ENGINEER OR TECHNICIAN with the appropriate tools. Failing this, return the board to your distributors for repairs.

Pushbutton Interface to Digital Port gives random readings.

Detailed Description:

I connected a push button to one of the port lines. When it is closed it is set to +5V (yield a logical 1) and the program reads a logical 1. However, when it is open, random numbers are read by the program on port A.

Solution:

1. **'PullDown' Resistor required when switch is in an open state.**

   When the push button is not connected, it seems that the port line is floating (not connected). You must connect a 'pulldown' resistor (experiment from 330R to 1k) to the port line in order to ensure that it is in a defined state. Also note that if a port is configured as an input then all unused lines MUST be grounded to Digital Ground.

Chapter 12: Appendix E: Waveview

Waveview

Data acquisition and analysis package

- **High speed streaming to disk**
- **Unique parallel board streaming**
- **Continuous buffer streaming**
- **Automatic use of XMS and EMS**
- **Digital storage oscilloscope**
- **Strip chart**
- **FFT's**
- **Waveform graphs**
- **Power spectrum graphs**
- **Infinite power and zoom**
- **Text, Epson and LaserJet III output**
- **Hercules, EGA and VGA support**
- **Runs on XT's and 100 MHz Pentium's**
- **Menu driven keyboard and mouse interface with online manual**

Introduction

Waveview is a powerful data acquisition and analysis package for the PC104-30 range of boards. It will run on any machine from an old XT to a 450 MHz Pentium III or Higher but still makes full use of EMS and XMS memory and disk space.

The boards supported are the PC26, PC104-30, PC39, PC104-30B, PC104-30C, PC104-30D, PC104-30DS, PC104-30DS4, PC104-30PGL, PC104-30PGH, PC104-30Fxx, PC104-30Gxx, PC126, PC127. Streaming is only possible on boards with DMA. The PC26, PC104-30, PC126 and PC127 do not have DMA. Waveview uses polled IO to sample from these boards.

Possible applications for Waveview include:

- 16 Channel digital storage scope
- Spectrum analyser
- Strip chart data logger
- High speed streamer (up to 48 channels)
- Continuous process monitoring

Data can be streamed to disk (using DMA) at the full 330 kHz throughput of the PC104-30F on a reasonably fast machine and disk. The number of samples acquired using streaming is limited only by disk space. Data can also be streamed to memory (using DMA). The number of samples is only limited by available conventional, EMS and XMS memory

Waveview supports the use of up to 3 boards simultaneously. Each board has its own configuration information, data sets, graphs and FFT's. Streaming data from multiple boards simultaneously is possible if each board is jumpered to use a different DMA level.

Using Waveview

Running Waveview

Make sure you are in the Waveview directory. If you are not type **cd\PC104-30\waveview** to get there. You can run Waveview by typing **wv** followed by any command line options you need. Table E - 1 shows possible command line options:

Table E - 1: Command Line Options

Option	Function
filename.cfg	Load single board configuration from filename.cfg. If no configuration file is specified Waveview will attempt to use ww.cfg.
filename.cfm	Load configuration for all 3 boards from filename.cfm.
/n	Do not install the DMA buffer. This saves memory when you are using a board that does not have DMA (such as the PC126).
/h	Display a summary of command line options.
/g3 or /g7 etc.	Use specified graphics card (3=EGA, 7=Hercules, 9=VGA) overriding Waveviews auto detection.
/gdriver,n	Use specified BGI driver in mode n. This allows you to use any SVGA and other graphics drivers you may have (e.g. /gsvga16,5).
/bs	Skip past the opening screens and start sampling data (for use in batch files).
/bm	Like /bs but starts multiboard sampling.

Example: **wv a.cfg /n**

This will run Waveview and load configuration information previously saved in a.cfg without installing the DMA buffer.

The User Interface

Waveview divides the screen into up to 4 graph windows. A menu bar is displayed above the windows and a graph command bar below. The board number is use is displayed in the top right hand corner of the screen. Menus are pulled down by pressing Alt and the highlighted character in the menu name or by clicking on the menu. For example pressing *Alt-F* will pull down the *File menu*.

Once a menu has been pulled down the cursor keys are used to move the selection bar up and down within the menu and to get to other menus on the menu bar. Menu options are selected by pressing *Enter* or *Space* while the selection bar is on the option. Context sensitive online help can be accessed by pressing *F1* at any time. Pressing *Shift-F1* brings up the help contents page.

The currently active graph window is displayed with a highlighted border. *Tab* and *Shift-Tab* or a mouse click in the desired graph window are used to move between graphs. Each graph has a number that is displayed in the top left hand corner of the graph. Each graph can display a different data set or power spectrum.

When a graph is displayed in the active graph window the graph command buttons are displayed along the bottom of the screen. Graph commands are accessed by pressed the highlighted character of the button or by clicking on the button. The cursor keys are used to pan along the graph.

Pressing *1, 2, 3, 4, 5, 6, 7, 8, 9 or 0* rapidly compresses the displayed waveform allowing more of it to be seen. This corresponds to the *compression* option on the *Display menu*. 15 different compression levels are selectable from this menu option. The highest level displays 16384 samples per pixel in the graph window. On a VGA display a full width graph window is 573 pixels wide so 9 338 032 samples can be displayed on screen. If the sampling frequency of the Waveform is 200 kHz this is equivalent to 46 seconds of data.

The mouse can be used to zoom in by holding down the left button and dragging out a rectangle and releasing the button. The same thing can be done by pressing Z on the keyboard and using the cursor keys to select the zoom rectangle. The compression feature can be used to get an overall view of the waveform - interesting parts can be expanded using the zoom and pan features.

Getting Started

Once you have run Waveview pull down the *Card menu* (press *Alt-C* or click on it). Select the *base address* option and fill on the base address of your board. Then fill in what type of board you are using the other settings (DMA, input ranges etc.) as jumpered on your board. Waveview uses the factory defaults to start with. On boards with software selectable ranges (such as the F and G series) Waveview will configure the board as specified on the menus.

When you have configured your board go to the *File menu* and select *Save set-up*. This will write the set-up information to ww.cfg (the default configuration file). This set-up file will be automatically loaded when you run Waveview in future.

Now go to the *Ad in menu* and try using the *Voltmeter* and *Begin sampling* functions. When you have sampled some data the waveform will be displayed in the active graph window. You can use the *Analyse menu* to calculate FFT's or the *File menu* to save the data to disk etc. Have a look through the online help to get an idea of what the menu options do.

Sampling Data

The Streamer

Waveview incorporates an advance streamer that can stream data from up to 3 boards simultaneously to disk and/or memory. The streamer requires DMA. See *Board Capabilities* at the end of this appendix for details on sampling from the PC126 and other non-DMA boards.

Streaming to Memory

The *destination* option on the *Ad in menu* must be set to memory. The *begin sampling* option will then invoke the streamer and transfer data from the card to memory. XMS and EMS memory is used automatically as required. If you have 3 MB of XMS and 1 MB of EMS available you can capture two million samples in one go. You should be able to achieve a throughput of 330 kHz even on a 386 class machine

Streaming to Disk

The *destination* option on the *Ad in menu* must be set to disk. You will be prompted from the name of the file to stream to. The begin sampling option will invoke the streamer and transfer data from the card directly to the specified disk file. The number of samples is only limited by available disk space.

Throughput of 330 kHz can be achieved on fast machines and disks but any disk caching software must be disabled. Programs such as SMARTDRV.EXE (supplied with DOS and Windows) buffer disk writes and write the data out when the machine is idle. This interferes with the streaming process. It is not necessary to disable such software when streaming at low speeds. Try streaming with the disk cache installed - if you get a buffer overrun error then it must be disabled.

Data streamed to disk is automatically attached (see the help on the *Attach data* option on the *File menu*). This allows you to view large waveforms without having to load the data into memory.

Before streaming starts Waveview expands the size of the destination file so that it is big enough to hold all the samples needed. This increases the maximum possible throughput as overwriting an existing file is quicker than creating a new file. The *Pad files* option on the *Options menu* can be used to turn this feature off if high speed streaming is not required.

Continuous Buffer Streaming

This option is controlled by the *Continuous buffer* option on the *Ad in menu*. When it is on and the number of samples required is reached, streaming continues and the oldest data is overwritten by new data. This continues indefinitely until escape is pressed and streaming is stopped. The data set (on disk or in memory) is treated as a large FIFO buffer.

You can use this option to monitor a process continuously. Streaming could be set up to take 3 million samples to a disk file at 50 kHz giving a sample time of 60 seconds. Streaming would continue indefinitely until something interesting happened. It could then be stopped and the previous 60 seconds of data analysed to find the source of the problem etc. etc. Waveview automatically adjusts the data set so that the first sample is the oldest sample in the data set.

When using continuous buffer streaming Waveview may increase the number of samples in the buffer slightly. If you ask for 10000 samples at 100 kHz you may get 10240 samples instead. The increase is greater at higher frequencies but is never more than 2048 samples.

Parallel Board Streaming

You can configure Waveview for multiple boards by selecting each board in turn from the *Board number* option on the *Card menu* or by pressing Shift and the number of the board you want to access. Each board has its own independent configuration, data sets and graphs. Each board must be jumpered for a different DMA level. You could use 3 PC104-30D's on DMA levels 5, 6 and 7 to create a 48 channel 600 kHz system.

Configure all the streaming details for each board individually (disk/memory, frequency, continuous buffer etc.) and check that each is working as required by sampling from each board on its own. Then use the *Multiboard sampling* option on the *Ad in menu* to select boards to sample and to start streaming. Streaming for each board will proceed in parallel with each board running at its own pace. Some of the boards can stream to memory and some to disk as needed all at different frequencies.

It is possible to achieve a combined throughput of 600 kHz on a 50 MHz 486 when streaming to memory from 3 PC104-30GA's. Generally it is difficult to get a throughput of more than 330 kHz when streaming to disk.

Program Transfer Sampling

Streaming is not possible on boards without DMA. Waveview uses polled IO (program transfer) techniques on these boards or on other boards when the DMA buffer has not been installed. The number of samples is limited to 32000 and fast machines (386 class) are required to achieve full throughput. On a 33 MHz 486 maximum throughput possible on the PC127 was about 90 kHz, just short of the boards maximum of 100 kHz. The maximum throughput of the PC126 is 50 kHz and this was easily achieved.

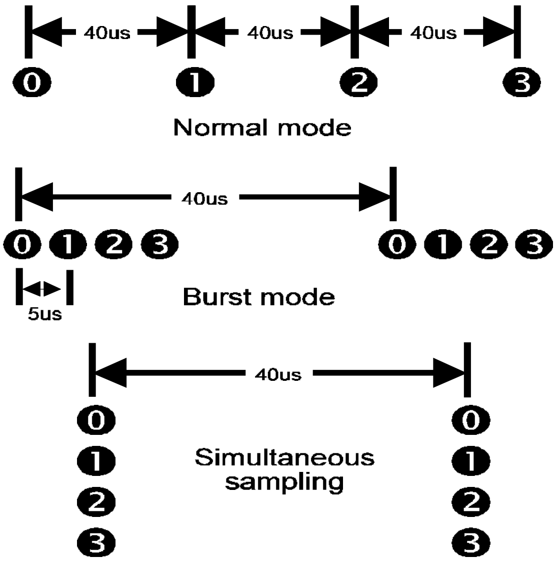
To prevent any loss of data while interrupts (for keyboard, real time clock etc) are being processed, Waveview disables all interrupts while doing program transfer sampling. This stops the real time clock so you will have to reset it from time to time.

Waveview still makes use of XMS and EMS memory for storing data sets sampled via program transfer once they have been sampled and for FFT results.

Burst Mode

Burst mode sampling is used to minimise the skew between channels sampled in the same block. When burst mode is enabled (*Ad mode* on the *Ad in menu*) all the selected channels are sampled at each clock pulse at the full throughput of the board. Simultaneous sampling takes place on boards with simultaneous sample and hold only in burst mode.

The illustration on the following page shows the spacing between samples taken from channels 0, 1, 2 and 3 on a PC104-30D at 25 kHz in normal and burst mode. The part of the diagram shows how the samples would be spaced if taken on a PC104-30DS which uses simultaneous sampling and hold in burst mode.



The Digital Storage Oscilloscope

The digital scope requires DMA. It samples up to 32000 samples from a selection of channels and displays the resulting waveform immediately. One of the channels can be use for software triggering (rising/falling edge in a specified voltage range) with pre, mid and post options. This can be used to continuously hold and display a waveform. It is best used with small numbers of samples (500 or so - just enough to show the waveform) so that screen updates are fast. Data sampled can be analysed and saved normally. It is accessed from the *Scope menu*.

The Strip Chart

The strip chart graphs the data as it is acquired on a sample by sample basis. The time needed to update the graph display means that it is only suitable for low sampling frequencies (generally less than 20 Hz). It is useful for long term monitoring of a process when you need to see the data as it is produced. It is started by selecting the *Strip chart* option on the *Ad in menu.*

The Voltmeter

The voltmeter continuously samples and displays the voltages on all or selected channels. It is useful for calibration and testing purposes. It is started by selecting the *Voltmeter* option on the *Ad in menu*.

Displaying Data

The is used to control how waveforms and power spectra are displayed. The settings shown on the menu always correspond to those of the

style, type of plot and time axis can all be changed from this menu.

Controlling the Time Axis on Waveform Graphs

sample in the waveform was captured relative to the first or it displays the actual time the sample was captured (based on the PC's clock). This is controlled by *Real time* option.

The normal graph only shows data or power spectra from one data set. The *Graph* *Display menu* is used to select a multiboard graph.

number (shown in the top left corner of the graph) from different boards into one graph.

Channels option on the is used to select boards and channels to display. This option will display a dialogue box showing all channels from all

that if the label for a channel is changed this change cannot be undone by clicking cancel or pressing escape to exit the dialogue box. The option on the *Display menu*
graph has a distinct colour. All the data sets for the graph must have the same frequency and number of samples.

of the same board at the same base address. Data can then be sampled normally from each copy and a multiboard graph used to compare the waveforms and

Digital IO and DAC Support

Waveview supports mode 0 (simple) digital IO. Individual decimal, hex and binary values can be written to and read from the digital output and input ports. Individual ports can be configured and input or output (the PC126 and PC127 have two 8 bit ports - one fixed as input and one as output). The digital ports are controlled from the *Digital menu*.

The voltage on each DAC can be set from the *Da out menu*.

Data Output, Printing and Plotting

Waveview can produce output on Epson compatible dot matrix printers, HP Plotters and the HP LaserJet III laser printer. Waveforms and power spectra can also be exported as ASCII text for easy importing into spreadsheet packages.

Exporting Data As Text

Use the *output* option on the *File menu* to select text output. You will be prompted for a file to write to and the current data set will be dumped as text. The delimiters used and other output options can be set from the *text* option on the *Options menu*. You can choose to output just the waveform and/or its power spectra, to write the actual binary value of each sample or the voltage, to put the data in columns etc. etc. Most programs will accept data in text files.

Printing on Epson Compatible Dot Matrix and HP Lasetjet III Printers

Screen dumps can be produced on Epson dot matrix printers and on the HP LaserJet III by selecting a printer from the *printer* option on the *Options menu* and then using the *print* option on the *File menu*. Waveview uses the extra resolution of the printer to dither the screen colours.

Plotting on HP Plotters Or the HP Laserjet III

High quality plotted output can be produced by selecting a plotter device from the plotter option on the Options menu and then using the plot option on the File menu. Unless your printer/plotter has a lot of memory it is best to just have one graph on the screen at a time when plotting. Waveview will attempt to plot all graphs on the screen on the same sheet of paper but the printer often runs out of memory.

Getting the Most Out of Waveview Using Non-DMA Boards

The PC26, PC104-30, PC126 and PC127 do not have DMA. As explained under *The Streamer* and *Board Capabilities*, this makes streaming impossible and limits the number of samples in a single block of data to 32000. However Waveview will still use XMS and EMS memory to store waveforms once they have been sampled and to store FFT's.

You can also make more memory available for sampling be releasing the DMA buffer (it is only used for boards with DMA). This is done by running Waveview with the /n option or by using the *free DMA buffer* option on the *Ad in menu*. Doing this will release about 128K of conventional memory.

Board Capabilities

Table E - 2 shows the capabilities of all the boards Waveview supports. Note than although some of the boards have dual channel DMA Waveview only needs one DMA channel.

Table E - 2: Board Capabilities

Board	DMA	Through-put	A/D chans	D/A chans	DIO lines	Burst mode	Sample & hold	Prog gains	Size
PC26	None	25 kHz	16	-	-	No	-	-	Full
PC-30	None	25 kHz	16	4	24	No	-	-	Full
PC39	Single	80 kHz	16	4	24	No	-	-	Full
PC-30B	Single	30 kHz	16	4	24	Yes	-	-	Full
PC-30C	Single	100 kHz	16	4	24	Yes	-	-	Full
PC-30D	Dual	200 kHz	16	4	24	Yes	-	-	Full
PC-30DS4	Dual	200 kHz	16	4	24	Yes	4 chans	-	Full
PC-30DS	Dual	200 kHz	16	4	24	Yes	16 chans	-	Full
PC-30PGL	Dual	200 kHz	16(8)*	4	24	Yes	-	1,10,100,1000	Full
PC-30PGH	Dual	200 kHz	16(8)*	4	24	Yes	-	1,2,4,8	Full
PC126	None	50 kHz	16	2	16	No	-	-	Half
PC127	None	100 kHz	4	2	16	Yes	4 chans	-	Half

Table E - 2: Board Capabilities

Board	DMA	Through-put	A/D chans	D/A chans	DIO lines	Burst mode	Sample & hold	Prog gains	Size
PC-30GA	Dual	100 kHz	16(8) ~	4	24	Yes	-	1,10,100,1000	2/3
PC-30G	Dual	100 kHz	16(8) ~	-	24	Yes	-	1,10,100,1000	2/3
PC-30GAS16	Dual	100 kHz	16(8) ~	4	24	Yes	16 chans	-	2/3
PC-30GS16	Dual	100 kHz	16(8) ~	-	24	Yes	16 chans	-	2/3
PC-30GAS4	Dual	100 kHz	16(8) ~	4	24	Yes	4 chans	-	2/3
PC-30GS4	Dual	100 kHz	16(8) ~	-	24	Yes	4 chans	-	2/3
PC-30FA	Dual	330 kHz	16(8) ~	4	24	Yes	-	1,10,100,1000	2/3
PC-30F	Dual	330 kHz	16(8) ~	-	24	Yes	-	1,10,100,1000	2/3
PC-30FAS16	Dual	330 kHz	16(8) ~	4	24	Yes	16 chans	-	2/3
PC-30FS16	Dual	330 kHz	16(8) ~	-	24	Yes	16 chans	-	2/3
PC-30FAS4	Dual	330 kHz	16(8) ~	4	24	Yes	4 chans	-	2/3
PC104-30G	Dual	100 kHz	16(8) ~	-	24	Yes	-	1,10,100,1000	PC104
PC104-30GA	Dual	100 kHz	16(8) ~	4	24	Yes	-	1,10,100,1000	PC104
PC104-30F	Dual	330 kHz	16(8) ~	-	24	Yes	-	1,10,100,1000	PC104

Table E - 2: Board Capabilities									
Board	DMA	Through-put	A/D chans	D/A chans	DIO lines	Burst mode	Sample & hold	Prog gains	Size
PC104-30FA	Dual	330 kHz	16(8) ~	4	24	Yes	-	1,10,100,1000	PC104

* The board can be jumpered for either 16 single ended inputs or 8 differential inputs.

~ The board can be software configured for either 16 single ended inputs or 8 differential inputs.

Chapter 13: Index

Index

A/D Calibration.......7-1
Calibration Procedure.......7-2
Connections.......7-2
Equipment Required.......7-2
Requirements.......7-1
Setting the Reference Voltage.......7-2
A/D Calibration Software
CAL30FG.EXE.......7-5
A/D Clock/Trigger
Default Setting.......3-16
External Clock.......3-16
Internal Clock.......3-16
Internal Clock to ADC and External Trigger.......3-16
Internal Clock/External Trigger.......3-16
A/D Configuration
A/D Clock/Trigger.......3-16
A/D Input Mode.......3-15
A/D Voltage Range Setting.......3-15
A/D Converter Codes.......6-1
A/D Strobes
Hardware Strobes.......2-3
Software Strobes.......2-3
A/D Subsystem
Clearing.......6-3
A/D Sub-system.......2-1
A/D Converter.......2-1
A/D full scale input ranges.......1-3
A/D Resolution.......1-3
A/D throughput rate.......1-3
FIFO buffer.......2-2
Input Multiplexer.......2-1
Non-linearity.......1-3
Number of A/D inputs.......1-3
Programmable Gain Amplifier.......2-1
Sample and Hold Unit.......2-1
A/D Voltage Range Setting.......3-15

Analogue I/O....4-7
Base Address....3-11
  DIP Switch Setting....3-2
Block Mode Triggering
  Block Trigger Mode....2-8
  Normal Mode....2-8
Board Initialisation....2-6
Bus Interface
  Decoding the Base Address....2-3
  Generation of DMA Signals....2-3
  Generation of Interrupts....2-3
CAL30FG.EXE....7-5
Calibration Procedure....7-2
Changing the Jumper/DIP Switch Setting....3-11
Channel Gain
  Setting....6-4
Channel List....2-10
Clearing the A/D Subsystem....6-3
Compatibility....9-1
Component Layouts....10-1
Configuration Registers
  A/D Configuration Register....5-39
  Clock Source Configuration Register....5-41
  D/A Configuration Register....5-41
Configuring the PC104-30 Board....3-1
  A/D Input Range....3-1
  Base Address....3-1
  Bus Interface....3-1
  D/A Output Range....3-1
  Uncommitted Counter/Timer....3-2, 3-17
Connection Guidelines....4-9
Counter/Timer Clock Source
  Clock Source is Grounded....3-18
  Pulses From the Clock Divider....3-18
  Pulses From the External Clock Line....3-18
  Pulses From the Master Clock....3-18
Counter/Timer Configuration
  Counter/Timer Clock Source....3-17
  Counter/Timer Enable....3-19
  External Clock Line....3-17
D/A Sub-system....2-1
DAC0 and DAC1
  Converter Codes....6-1

DAC0 to DAC3
D/A non-linearity........1-3
D/A resolution........1-3
D/A throughput rate........1-3
Full scale output ranges........1-3
DAC0 to DAC3 Calibration........7-5
DAC2 and DAC3
Converter Codes........6-2
Data Transfer from A/D to Memory
Simple Polled I/O........2-6
Single Block DMA........2-7
Diagnosing Faults........11-1
Digital I/O Section........2-10
Mode 0 (Basic Input/Output)........2-10
Mode 1 (Strobed Input/Output)........2-11
Mode 2 (Strobed Bi-directional Input/Output)........2-11
Digital I/O Section Operating Modes
Mode 0 (Basic Input/Output)........4-10
Mode 1 (Strobed Input/Output)........4-11
Mode 2 (Strobed Bi-directional Input/Output)........4-13
DIP Switch Setting........3-2
Direct Memory Access
Multi Block DMA........2-7
Single Block DMA........2-7
DMA
General Description........1-7
DMA Channel Allocations
Default Setting........3-14
DMA Data Format........6-11
DMA Jumper Settings
Default Setting........3-14
EDR Features........1-5
Error Detection........6-12
Extended Memory........6-11
Fault Finding........11-1
Features of the PC104-30
A/D Sub-system........1-2
D/A Sub-system........1-3
Digital I/O Sub-system........1-4
Interface logic........1-4
Programmable Gain........1-2
Handshake Signals
Input Operations........4-12
Output Operations........4-12

Hardware Specifications....8-1
Interconnections....4-1
  To the IBM Backplane....4-1
  User Interface....4-1
Interrupt Level Allocations....3-11
Interrupt Source Configuration
  End of Conversion....3-13
  End of DMA Block....3-13
  On Each Pulse From the Counter/Timer....3-13
Layout Diagram....10-1
Mode 0 Characteristics....2-10
Mode 1 Characteristics....2-11
Mode 1 Handshake Signals....4-12
Mode 2 Characteristics....2-11
Mode 2 Handshake Signals....4-14
PC104-30
  A/D Sub-system....1-2
  Accessories....1-8
  Address Locations....5-1
  Counter/Timer....2-5
  D/A Sub-system....1-3
  Digital I/O Section....4-10
  Digital I/O Sub-system....1-4
  Initialisation Procedure....6-2
  Interface logic....1-4
  Interrupt Source....3-13
  Programmable Gain....1-2
  Range of Boards....1-1
  Register Layout....5-2
  Starting up....1-7
  Versions....1-1
PC104-30F, FA, G, GA
  Differential Inputs....3-15
Program Transfer....1-7
Programming Guide....6-1
Programming the PC104-30....6-1
  Clearing the A/D Subsystem....6-3
  Converting From Binary To Analogue Values....6-1
  Digital I/O....6-4
  Dual Channel Gap-free DMA....6-9
  End of DMA Block Interrupt....6-12
  Initialisation....6-2
  Interrupts....6-7

- Introduction........6-1
- Loading the Channel List/Block Counter........6-6
- Obtaining a Series of A/D Conversions by Polled I/O........6-7
- Obtaining a Single A/D Reading........6-5
- Setting Channel Gain........6-4
- Setting the Sample Rate........6-5
- Single Channel DMA........6-8
- Writing to the D/A Converters........6-4

Recommended Analogue Input Schemes........4-7

Register Configuration
- DMA/IRQ Config Register........5-25, 5-27

Register Structure
- A/D Clock Divider Register........5-13
- A/D Clock Prescaler Register........5-12
- A/D Control/Channel Register........5-7
- A/D Data Low Byte........5-4
- A/D Data/Status Register........5-5
- A/D Mode Register........5-8
- Block Counter........5-4
- DAC0 Register (Low Byte)........5-23
- DAC0 Register High Byte........5-24
- DAC1 Register (Low Byte)........5-24
- DAC1 Register High Byte........5-25
- DAC2 Register (Low Byte)........5-31
- DAC2 Register High Byte........5-31
- DAC3 Register (Low Byte)........5-31
- DAC3 Register High Byte........5-32
- Digital I/O Control........5-21
- Digital I/O Port 0........5-19
- Digital I/O Port 1........5-19
- Digital I/O Port 2........5-20
- Gain Memory 0 Register........5-35
- Gain Memory 1 Register........5-36
- Gain Memory 2 Register........5-37
- Gain Memory 3 Register........5-38
- Gain Read Back and Board Type Register........5-32
- Timer Control Register........5-15
- User Counter Register........5-14

Selection of DMA Levels........3-14

Signal Definitions
- +5V........4-5
- Analogue Ground........4-2
- CH0 - CH15........4-2
- DAC0 Output........4-3

DAC1 Output 4-3
DAC2 Output 4-3
DAC3 Output 4-3
Digital Ground....4-6
External Clock....4-5
External Trigger 4-5
Port A0 - A7....4-5
Port B0 - B7 4-5
Port C0 - C7....4-5
Software Support
EDR Features....1-5
Visual Basic Custom Controls Features 1-6
Waveview Features 1-6
Specifications....8-1
Timing and Control
Clock Divider....2-4
Clock Selection Multiplexer 2-4
Counter/Timer....2-5
Crystal Oscillator....2-4
Troubleshooting 11-1
Uncommitted Counter/Timer 3-17
User Connection
Available Signals.... 4-1, 4-4
Wait States....3-12
Waveview
Board Capabilities....12-11
Data Output, Printing and Plotting 12-10
Digital IO and DAC Support 12-9
Displaying Data....12-9
Getting Started 12-4
Getting the Most Out of Waveview Using Non-DMA Boards 12-10
Introduction....12-2
Sampling Data....12-4
User Interface....12-3
Using....12-2
Waveview Features 1-6
Writing to the D/A Converters....6-4